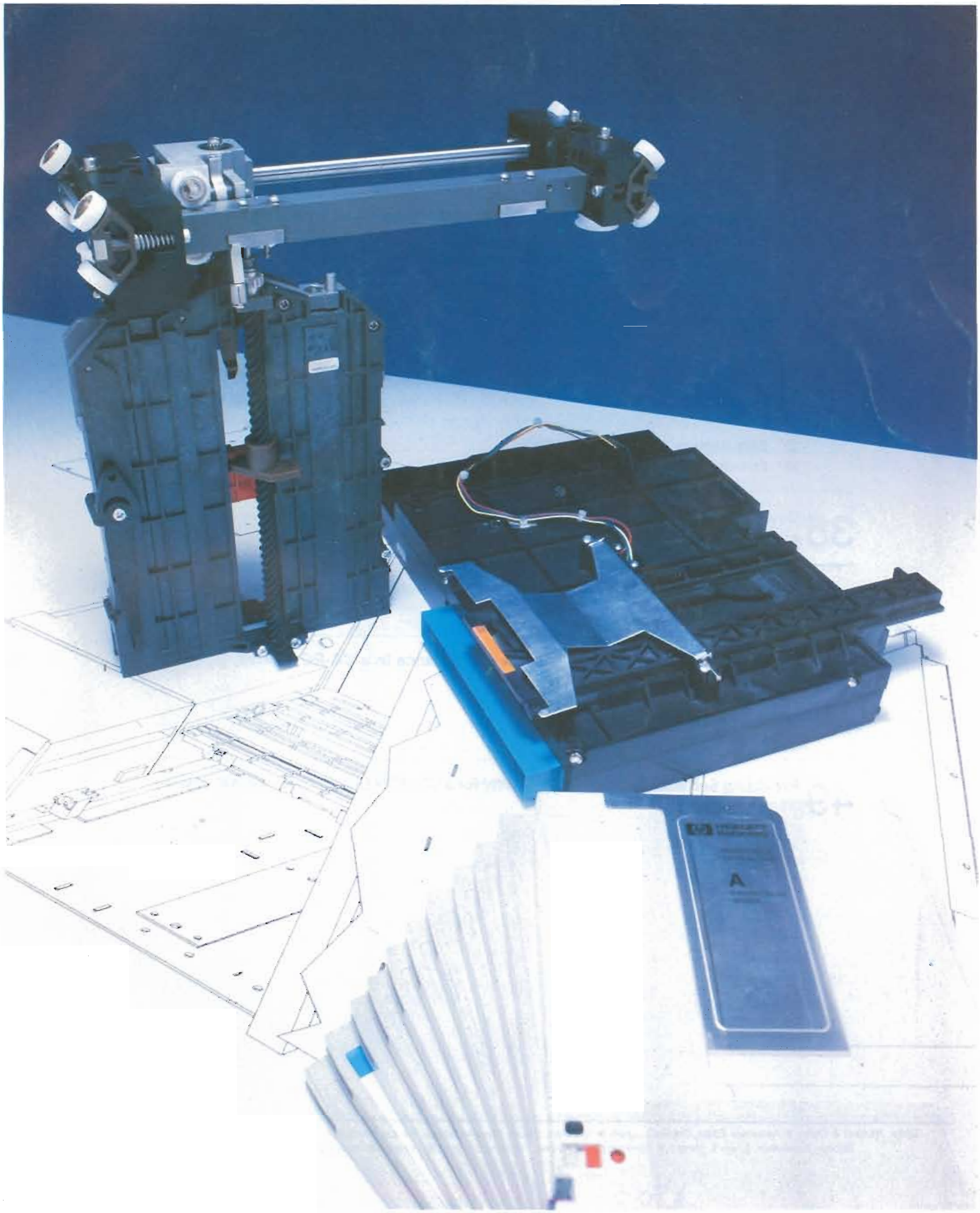
HEWLETT-PACKARD JOURNAL

DECEMBER 1990



HEWLETT
PACKARD

A Rewritable Optical Disk Library System for Direct Access Secondary Storage

This autochanger system can store up to 20.8 Gbytes of data on-line. Applications include archival storage, automated backup and recovery, and document storage and retrieval.

by Donald J. Stavely, Mark E. Wanger, and Kraig A. Proehl

HEWLETT-PACKARD MANUFACTURES a wide range of computer peripherals. Customers for these peripherals include not only users of HP systems, but also OEM customers and others who use HP peripherals with non-HP host systems.

Supplying peripherals to OEM customers has been a major initiative for Hewlett-Packard and has had a large impact on how we plan and evolve our business strategies. To be successful in the OEM business has required that we develop a broader and more timely understanding of the market than we had in the past. We feel that our experience as a system company gives us valuable insights into how our peripherals work in systems and applications to solve real customer needs.

HP's Greeley Storage Division is responsible for high-end secondary storage devices that are used for backup and archival storage on computers, mainframes, and networks of workstations. Our current product offering is a family of low-cost, autoloading, streaming, 1/2-inch GCR tape drives.[1]

As we looked to the future, we naturally focused our attention on advances in tape technology. Emerging products were using air bearings for media reliability, a thin-film 18-track head for very high transfer rate, and a compact tape cartridge for ease of handling. Initially, this technology seemed a good match to what our current HP and OEM customers needed. Customers were asking for faster backup to reduce planned system downtime—or more accurately, to keep from increasing their downtime as their disk storage requirements grew. They also need ever higher levels of reliability to minimize unplanned system downtime.

Unfortunately, simplistic market research—asking customers what they want—often yields only predictable and simplistic answers. They want what they have now, only faster, cheaper, more reliable, and so on. In other words, customers may be too close to their problems to see them from a new perspective.

We evolved a much more powerful market research process that consists of three steps. The first step is to gain a thorough knowledge of how customers do business. What applications do they run? How much disk space do they have? How do they do backup today? What else do they use tape drives for?

The second step of the process is to try to solve customers' problems in the abstract—matching available technologies to a high-level model of each customer's business. The last step is to present a coherent vision of the future back to our customers. In essence, we are trying to help them look past the limitations of today's solutions and help them architect the solutions of tomorrow. We call this developing an "imaginative understanding of user needs."

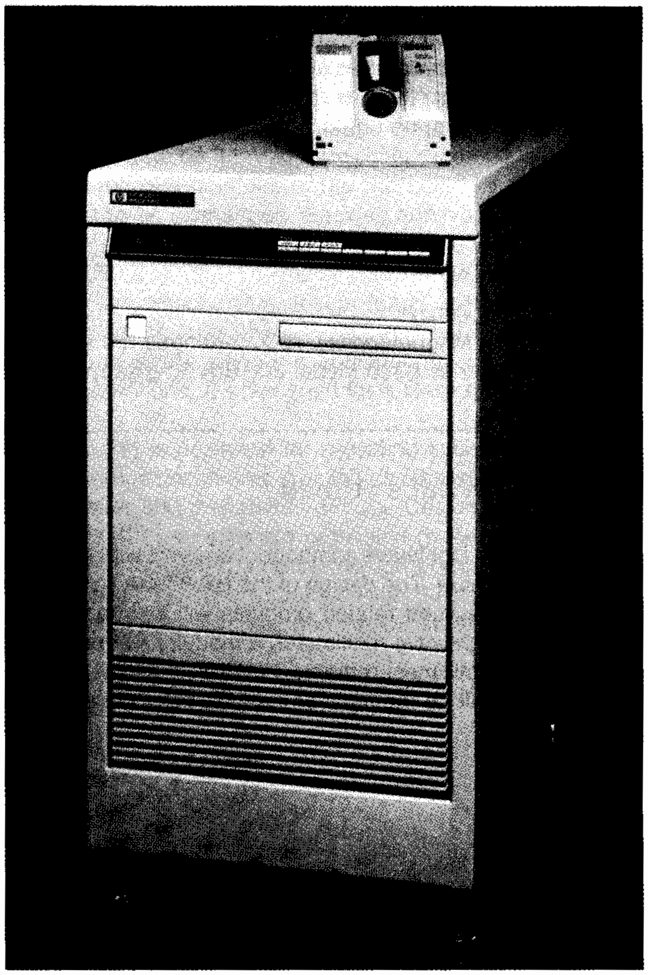


Fig. 1. *The HP Series 6300 Model 20GB/A rewritable optical disk library system stores up to 20.8 Gbytes of data on 32 optical disks. An autochanger automatically selects the correct disk and inserts it into one of two internal drives.*

In probing more deeply with both HP system customers and OEM customers, we found that their responses to our simplistic market research were indeed conditioned by the properties and limitations of tape technology. The truth is that customers don't like tapes. Tapes are inherently off-line devices requiring sequential access to data and operator intervention to handle the media. What customers really want is direct access to all of their archival data, without special utilities and without operator intervention. We call this concept direct access secondary storage, or DASS. When we clearly articulated this concept and fed it back to customers, we received consistently positive responses.

Rewritable optical disk technology, configured in an automated library, has exactly the right properties to meet this customer need for direct access to archival data. Optical disks are removable, rugged, reliable, and fairly inexpensive on a cost-per-megabyte basis. Transfer rates are competitive with many current tape products. And because it is a disk drive, an optical disk drive attaches to the host system using standard disk drivers and file systems. This can give direct, transparent access from current applications without modification. The connection between optical disks and secondary storage makes perfect sense, but it was not obvious to either customers or the optical drive vendors themselves.

The optical disk autochanger plays the other key role in the DASS concept. With many gigabytes of on-line Winchester-disk storage, a typical host system requires tens or even hundreds of gigabytes of secondary storage for backup and archival information. In the DASS concept, this secondary storage must be on-line—accessible without operator intervention or special recovery utilities. Rewritable optical drives in an autochanger configuration provide a cost-effective answer to the customer need for direct access to huge amounts of historical data.

Reliability is the single most important attribute of an autochanger. The customer perception is that autochangers are "mechanical nightmares" that are fascinating to watch at trade shows but frightening to consider as a vital link in a company's computer operations. It was for this reason that Hewlett-Packard chose to design and build its own autochanger mechanism.

The philosophy used to guide the development was that reliability should not be tested into a product, nor even designed in—it must be architected in. An architecture that minimizes complexity, followed by careful design and rigorous testing, is the only way to achieve a quantum leap in reliability.

Optical Disc Library System

The result of these considerations is the HP Series 6300 Model 20GB/A rewritable optical disk library system, Fig. 1. The Model 20GB/A combines the convenience and low storage cost of optical-disk technology with the massive capacity of a library system to provide on-line access to vast amounts of infrequently accessed information. The Model 20GB/A is a direct access secondary storage (DASS) device that fills the price/performance gap between high-performance hard disks and low-cost tape storage (Fig. 2). Because of its huge, 20.8-Gbyte storage capacity and low cost per megabyte (Fig. 3), the product makes it feasible to store information on-line that has traditionally been stored off-line, and to automate labor-intensive backup and recovery processes. It also greatly reduces the floor space required for archiving (Fig. 4).

The Model 20GB/A uses magnetooptical technology (see box, page 8). Data is stored on removable 5¼-inch disks. Optical disks are not susceptible to head crashes and are much more tolerant of magnetic interference than magnetic media. Fingerprints and small scratches have no effect on the data. Data can last over ten years without the retensioning or reconditioning that tapes require.

The Model 20GB/A consists of an autochanger, two magnetooptical disk drives, and 32 5¼-inch, 650-Mbyte optical disk cartridges in a deskside cabinet. A mailslot is provided for loading or removing disks. The autochanger automatically selects the appropriate cartridge and inserts it into

(continued on page 10)

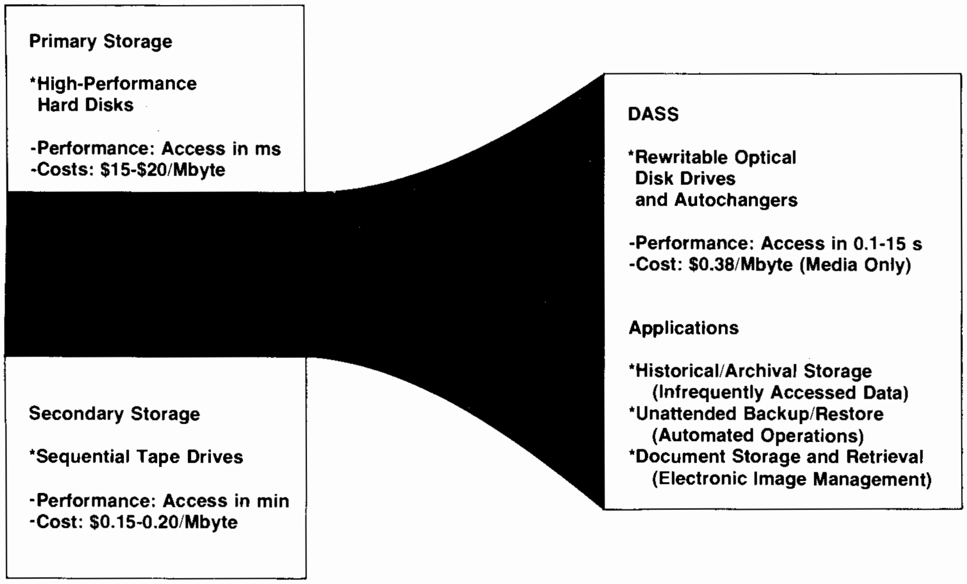


Fig. 2. *The optical disk library system is a direct access secondary storage (DASS) device, lower in cost than high-performance disk drives, but higher in performance than low-cost tape.*

Magnetooptical Recording Technology

Rewritable optical technology today encompasses three different methods:

- Magnetooptical
- Dye-Polymer
- Phase-Change.

The most durable and predominant technique in the market today is magnetooptical. This discussion will be limited to this technique, which is the method that Hewlett-Packard has chosen for introducing rewritable optical technology for direct access secondary storage.

Magnetooptical technology relies on the storage of information on a thin film of magnetic material. Like conventional magnetic recording, the information is stored on the media in the form of magnetic domains. The domains are aligned vertically, in contrast to most magnetic recordings today, which are based upon longitudinal magnetization. The important and significant difference comes from the fact that the processes of writing, erasing, and reading are performed with a light beam derived from a solid-state laser and associated optics, not by mechanical heads that come into contact or near contact with the recording surface. This attribute allows optical recording to have longer life and higher reliability than tapes and flexible disks. Optical disks are immune to the typical wearout modes that occur with contact or close proximity recording.

Recording

Thermomagnetic writing is the term used to describe the process of writing information on a thin magnetooptical film. The laser beam heat-modulates the magnetic film about its Curie temperature. The Curie temperature of a magnetic material is the temperature at which the material loses its coercive magnetic field. This occurs between 150°C and 200°C for typical magnetooptical thin films. When this occurs, the material loses all memory of its prior magnetization and can acquire a new magnetization as it cools in the presence of an external magnetic field.

The writing process is shown schematically in Fig. 1. The recorded information is stored on the magnetic medium by reversing a magnetic domain to store a one and by not reversing a domain to store a zero. Thus the precondition for writing information is for all domains to be initialized to the zero state. This means that, to overwrite data, an erase pass must be performed before the write pass to set up this initial condition of all-zero domain alignment. During the erase pass, the laser is turned on to heat the magnetic domains and an external magnetic field is applied in the proper orientation to change all of the domains to the zero state.

Data can be written on the erased track during a subsequent disk rotation. With the polarity of the external magnetic field reversed, the laser is turned on and off to heat only those domains that are to be changed to the one state. The external magnetic field required to erase or write data is supplied by a bias magnet which is typically positioned on the opposite side of the film surface from the optical head. This external bias magnet must have the ability to change magnetic polarity; therefore, it is typically an electromagnet or a permanent magnet that can be mechanically rotated to accomplish polarity changes.

When magnetooptical films are at room temperature, they typically exhibit coercivities of several thousands of Oersteds. This means that in the absence of laser heating, the magnetic field required to affect their state of magnetization is extremely large. Because of these high coercivities at operating and storage temperatures, magnetooptical records are less susceptible to damage from external fields than records on conventional magnetic storage materials such as those on flexible and rigid magnetic disks.

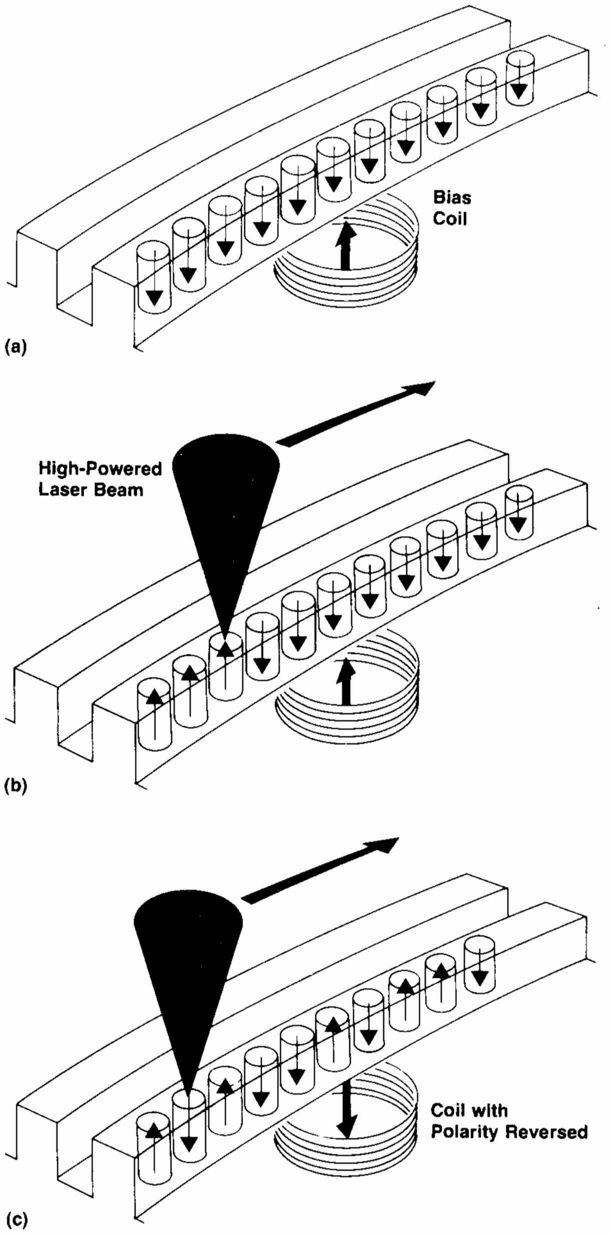


Fig. 1. *The magnetooptical write process. (a) All of the magnetic domains are magnetized north-pole-down. This all-zeros state is the precondition for writing. (b) The laser beam turns on for each domain that is to store a one. Heating the domain above the Curie temperature causes it to lose its previous magnetization and orient itself with the external magnetic field of the bias coil. (c) To erase the data, the polarity of the external field is reversed and the laser is turned on, returning all of the magnetic domains to the condition shown in (a).*

Readback

For data readout, information is extracted from the magnetooptical film by reflecting a polarized light beam off the magnetic film surface and detecting a change in the angle of polarization of the reflected beam. This physical phenomenon, upon which the magnetooptical rewritable technology is based, is known as the Kerr effect. It is manifested as a change in the state of polarization of light upon interaction with a magnetized medium. The amount of polarization rotation is small (less than one degree) but techniques used in film manufacturing can enhance the effect. In addition, a variety of detection and readout techniques have been developed to enhance the magnetooptical signal. As a result, good signal-to-noise ratios of 60 dB or more can be achieved.

Another magnetooptical readout alternative is based upon the Faraday effect. This effect is similar to the Kerr effect but relies on light transmitted through magnetic films. The interaction of the light with the film causes polarization state changes. This technique is not employed in the magnetooptical rewritable process primarily because of the low transmissibility of magnetooptical films and the difficulty of placing interactive optics on both sides of the media.

Magnetooptical Materials

Magnetooptical materials are composed of a rare earth element and a transition metal. Typical rare earth elements used in magnetooptical recording include gadolinium (Gd, $z = 64$) and terbium (Tb, $z = 65$). These rare earth elements are also called lanthanides. These elements are soft, gray metals that have good conductivity. As a group, the lanthanides are not very abundant. The most common lanthanide is cerium, which makes up only 3×10^{-4} percent of the mass of the earth's crust. The transition metals commonly used in magnetooptical recording include iron (Fe, $z = 26$) and cobalt (Co, $z = 27$). These elements contribute characteristics such as high melting temperature, good conductivity, and fairly high hardness. Alloys of rare earths and transition metals are amorphous and have been processed to achieve a high level of chemical stability. The transition metal provides the dominant magnetooptical interaction (Kerr effect) while the rare earth element helps to provide high vertical magnetic anisotropy.

Curie and Compensation Temperatures

The important parameters in processing magnetooptical films are the Curie temperature of the alloy (mentioned earlier) and the compensation temperature. The compensation temperature is the temperature at which the magnetization component of the transition element is equal and opposite to that of the rare earth element, so that the net magnetization is zero. The compensation point can be either above or below the ambient temperature. At the compensation temperature, since there is no net magnetization, the material cannot interact with external fields. Therefore, the coercivity is extremely high and the magnetic domains are very stable. For practical magnetooptical recording films in use today, the compensation temperature is kept well below the Curie temperature and the lowest operating temperature the film will see. The reason is that the interaction of the compensation point magnetic behavior can affect the requirements for Curie temperature recording. If the compensation temperature is in the region of operation, the magnetic properties change dramatically and can interfere with the designed magnetooptical recording process.

The compensation point for magnetooptical films is determined by the percentage of the rare earth element in the film. Typical percentages, for example, are for terbium to be below 19 to 20 atomic percent to keep the compensation temperature below ambient. If the percentage of terbium is higher, say above 22 to 23 atomic percent, then the compensation temperature can exceed ambient. At percentages greater than 27 or 28 atomic percent, there is no compensation temperature because the compensation temperature exceeds the Curie temperature, and magnetic properties above the Curie temperature dominate the film behavior.

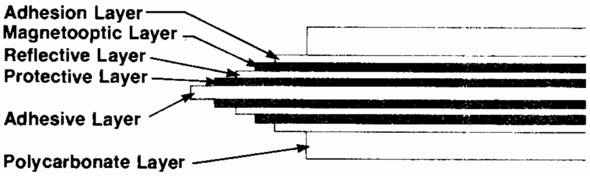


Fig. 2. *Magnetooptical disk construction.*

The Curie temperature is selected so that the laser light source can easily raise the magnetooptical film material to this temperature without exceeding the design limits on the laser power. A film with a lower Curie temperature requires less heat and therefore is more sensitive. Hence the Curie temperature controls the media sensitivity.

The Curie temperature for magnetooptical films is determined by the selection of the transition metal component. One way to control the Curie temperature is to adjust the ratio of cobalt to iron in the transition metal. As the ratio of cobalt to iron is increased, the Curie temperature is increased and the film sensitivity is decreased—more power is required to reach the Curie temperature.

Manufacturing

The manufacturing processes for magnetooptical disks have to take into account a wide variety of parameters. Important considerations include the following:

- Mechanical stability of the substrate, which is typically plastic (polycarbonate). Glass and aluminum have also been used. Some of the parameters of concern are warp, tilt, axial and radial runouts, and accelerations.
- Birefringence of the substrate, a condition in which the index of refraction is dependent on the polarization of the light.
- Dust protection. This is provided by a transparent layer that keeps dust, scratches, or other optical disturbances away from the focal plane of the recording surface.
- Surface reflectance control. This requires control of layer thicknesses and refractive indexes.
- Thermal characteristics, including thermal properties of films and surrounding structures and materials.
- Magnetic properties, which are determined by film composition and thicknesses.
- Protective coatings, such as dielectric barrier films for corrosion protection.

A typical cross section of a magnetooptical disk is shown in Fig. 2. The disk consists of two ten-nanometer-thick layers of magnetooptical film—one for each side of the disk—sandwiched between two polycarbonate disks. Dielectric material and adhesives separate and bond the layers.

Ed Sponheimer
Project Manager
Greeley Storage Division

one of the two internal drives. Operation is transparent to users, who see only a slightly slower response time when accessing optically stored data—approximately 100 milliseconds if the disk is already in a drive, or about 10 to 15 seconds if disks need to be exchanged. The HP-UX operating system recognizes each disk side as a 325-Mbyte mountable file system, so data access and software compatibility are the same as if the library system were a (slower) hard disk.

The Model 20GB/A conforms to ANSI and ISO specifications for continuous composite format 5¼-inch rewritable optical disks. This ensures compatibility with the HP Series 6300 Model 650/A stand-alone rewritable optical disk drive and the drives and media of other manufacturers. The system implements the Small Computer System Interface (SCSI) in asynchronous mode with separate IDs for both drives and the autochanger. The product is supported on HP 9000 Series 300 and 800 computer systems running the HP-UX 8.0 operating system.

Design Philosophy

The design criteria for the HP Series 6300 Model 20GB/A rewritable optical disk library system were focused on three points that we felt were essential to the successful launch of a peripheral with new functionality: minimizing time to market, making the product very reliable, and completing full system integration. The major design and architecture philosophies we used were high leverage of existing successful designs, design simplicity, and modular design with limited coupling between modules.

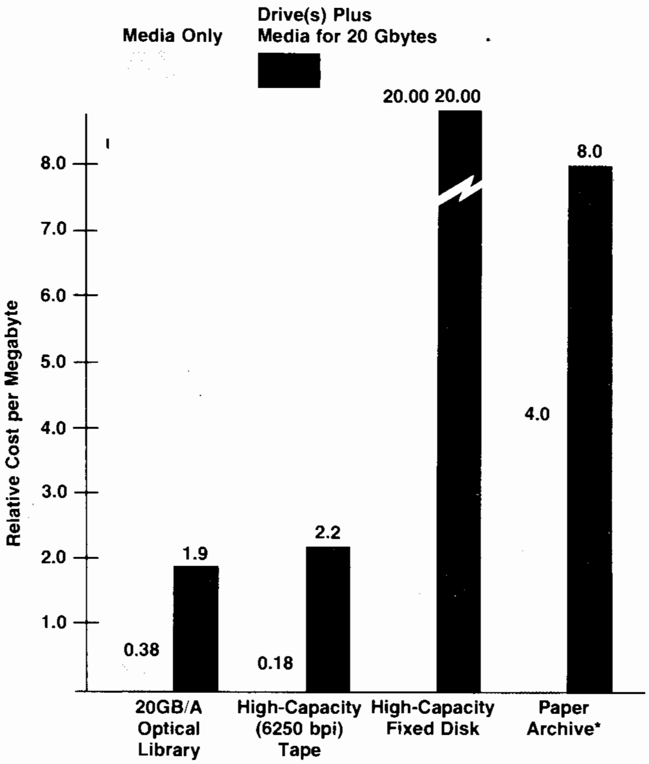
Leveraged designs allowed us to produce early breadboards rapidly with a high level of sophistication. In using leveraged designs, we also gained all the benefits of the engineer-years of effort that went into reliability engineering. We leveraged coupled servo architecture, motor/encoder design, a proprietary HP motor control IC, pulley and belt design practices and capabilities, and carriage and way design. We limited ourselves to off-the-shelf power supplies to decrease risk and tooling expense.

The design for reliability began with a study of failure rates on HP plotter products and the HP quarter-inch tape cartridge autochanger. We found that the predominant failures were associated with sensors, switches, motors, and solenoids. We next generated graphs of annual failure rate as a function of the number of sensors and as a function of the number of actuators. The architecture that followed from this analysis called for a minimum number of sensors and motors, and for a "passive payload" design, which means that no sensors or actuators were allowed on any moving parts. This strategy decreased the number of high-failure-rate parts and supporting parts as well, such as cables, connectors, and flexible circuits. It also eliminated flexing wires, which have fatigue problems. We were well aware that as printers have evolved in reliability, the hardest remaining reliability problem is the flexible cable going to the printhead.

The mechanical design of the autochanger is described in the article on page 14.

Another feature of the design is the servomechanism "sense of touch," which is tied directly to the passive

(continued on page 12)



*Varies with application. Conservative estimates are 0.01 per page and 0.02 per page with storage and retrieval.

Fig. 3. *Storage cost per megabyte for various alternatves.*

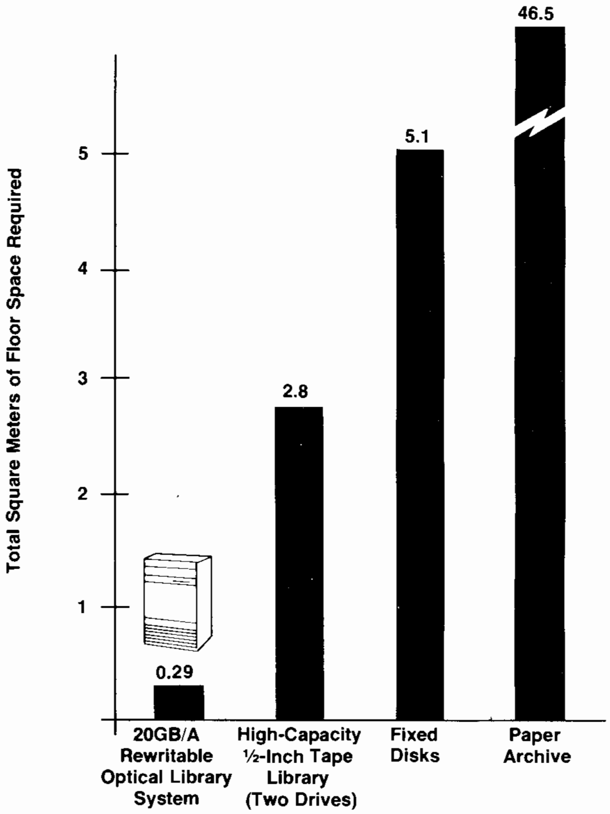


Fig. 4. *Storage space comparison for 20.8 gigabytes.*

Integrating the Optical Library Unit into the HP-UX Operating System

The HP Series 6300 Model 20GB/A rewritable optical disk library autochanger is unlike any other peripheral supported on the HP-UX operating system and therefore required a new approach to integration into the operating system. On one hand, its random-access attributes suggest a connection with the operating system that is disk-like. On the other hand, its need to share multiple disks with one or two drives hints at something that may need specific user or application support.

Although the design options were unrestricted, we wanted an integration method that would satisfy two overriding goals. First, the integration method should hide as much as possible the requirement to swap disks into and out of the drives. This transparency goal means that no special programs or utilities are required to access information on the autochanger. This holds for commands that rely on network services as well as commands that treat mass storage peripherals as raw devices. The second goal was that the integration method should have minimum impact (complexity, coupling, etc.) on the HP-UX operating system.

Design Choices

The accepted way of integrating WORM (write once, read many) autochangers relies heavily on the application. An application is provided with low-level control of the changer mechanism and low-level control of the drives. The application is responsible for swapping disks in and out of drives and tracking the location of disks. This method clearly does not allow transparency, but our solution needed to support this low-level control so that existing applications that already rely on it could be ported to HP-UX and so that other autochanger-specific utilities could be developed. The HP-UX **ioctl** system call is used to support these low-level commands.

A tempting way to model the integration is to view the entire autochanger as one large disk. This solution implies that the disk cartridges that make up this large disk travel as a group and remain in some logical order. Since disks in the autochanger are inherently removable, the administrative problems of keeping sets of disks together led us away from this solution.

The solution we settled on treats each side of a disk as an individual disk.[1] The system administrator is free to create file systems on these disks and mount them or access them in the raw mode using existing system calls such as **read()**, **write()**, and other utilities that use raw devices. A file system residing on an individual disk surface can be mounted anywhere in the directory structure of the HP-UX file system.

Neither existing commands nor application programs require modification to maintain their functionality. The file systems residing on cartridges maintain their NFS (Sun Microsystems' Network File System) functionality with other machines on the network. The file systems are also protected from power failure to avoid lengthy file system recovery processes.

By confining most of the changes to a driver, the goal of minimizing HP-UX changes was met. Fig. 1 illustrates the structure of the autochanger driver. The autochanger driver consists of two main parts: the surface driver and the changer driver. The existing disk driver is used to control the drives. This is the same driver that controls the stand-alone rewritable optical drive, the HP Series 6300 Model 650/A.

The changer driver provides low-level control of the autochanger mechanism. It accepts commands to move the disk in slot x to drive 2, to report whether there is a disk in slot 2, and similar tasks. The surface driver controls the swapping of disks and routes disk requests to the disk driver when the requested disk is in a drive.

The Swapping Algorithm

When requests for different surfaces occur, only one of those surfaces can be inserted into each drive. The other requests must be suspended. To avoid having these requests wait forever, we set a limit, called the hog time, on the time that a cartridge can be in a drive processing requests while other requests are waiting. Once this time expires, that cartridge is removed and the request waiting the longest is inserted. We have found that the hog time should be somewhat larger than the time required to exchange a cartridge. Twenty seconds has proved sufficient.

If a cartridge in a drive were to be replaced immediately when there are no additional requests for that surface, it is possible that shortly after the cartridge exchange is started, a request for the original surface could arrive. This could result in a cartridge swap for every request. To avoid this problem an additional limit called the wait time was added. This is the maximum time that a cartridge can reside in a drive without processing any requests while other requests are waiting to use the drive. Choosing the wait time too large increases the effective swap time of the autochanger. Making the wait time too small increases the chances of thrashing as a result of consecutive requests for the same surface. We found a wait time of one second to be sufficient to avoid the extra swapping in most instances.

Because we realize that certain configurations will require different hog and wait times, these values are configurable (via drive **ioctl** calls) while the system is running. For example, in a backup application, the hog time should probably be high, so that other processes won't seriously degrade the throughput. If an application program reads blocks of data and then processes that data for more than a second before it reads more data, the wait time should be increased to avoid swapping between reads.

Autochanger Driver Design

The design of the autochanger driver overcomes three problems:

- It avoids a file system check (the **fsck** command) of all the cartridges after a power loss.
- It avoids excessive swapping caused by the syncer.
- It supports the concept of asynchronous read and write operations.

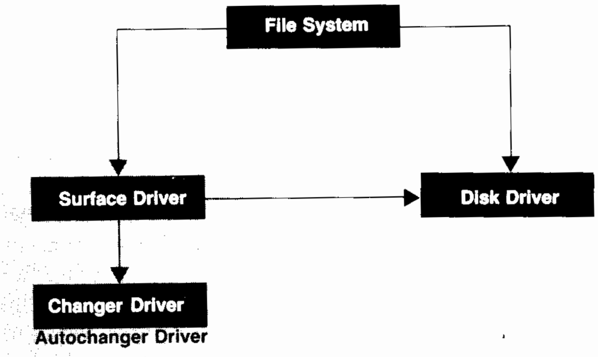


Fig. 1. *The autochanger driver consists of a surface driver and a changer driver. The existing disk driver is used to control the optical drives.*

Avoiding a Lengthy fsck. The superuser must execute the mount command for each surface to be accessed before users can perform file system operations to the autochanger. Normal Winchester disks require a file system check if the power cycles while the disk is mounted. Since we do not want to require the user to check every surface, which could take several hours, we make sure the file system on every disk is in a known, valid state before the disk is removed from a drive.

To do this, the autochanger driver uses the concept of virtually mounted devices. A virtually mounted device is not vulnerable to power failure corruption of its file system. When a device is virtually mounted, it is not directly connected to the rest of the file system. Only basic information about the file system and device is stored. The file system is available for use, but files cannot be accessed until the device has been physically mounted.

Cartridges in the autochanger that are waiting in their slots are only virtually mounted. When a file on a virtually mounted device is referenced by a user, the file system code uses the stored information to mount the device physically before the file is accessed. When the operations on that device are finished, the file system physically unmounts the device. This causes any buffers in the host that were not written to the device to be flushed. The file system is now in a state that would not require file system repair on a power failure. Cartridges only need to be virtually mounted when the system is first booted.

Avoiding Extra Syncer Swaps. To limit the number of modified buffers that haven't been written to the disk at the time a power failure occurs, there is a process that executes periodically that flushes these buffers. This process, called the syncer, schedules all the modified buffers to the drivers so they will be written to the disk. If a surface is removed from the drive without flushing all the modified buffers, the syncer will execute and require the cartridge to be reinserted. Having several cartridges mounted can cause excessive thrashing.

The solution is to write all the modified buffers for a surface before the cartridge is removed. This is done as part of the physical unmounting process.

Supporting Asynchronous Operations. Every block device driver in the HP-UX kernel must be able to support the concept of asynchronous requests. An asynchronous request essentially means that when the file system makes a request to the drive to perform say, a write to disk, the driver should queue the request and immediately return without actually doing the I/O. This tends to pose a problem in that, once the request is queued, some thread of execution must eventually complete the request. The normal method of doing this is through hardware-generated interrupts to the driver. However, this interrupt structure is absent in the autochanger driver.

There are two basic ways to solve this problem. One is to provide the extra thread of execution through the disk driver's interrupt calls. This adds extra complexity and coupling to both of the drivers. The method we have chosen is to create extra kernel daemons to provide the extra threads of execution necessary. Two daemons are used. A transport daemon is responsible for moving the cartridges, and a spinup daemon is responsible for spinning up the drive. The two daemons make it possible to overlap moves and spinups. For example, after a cartridge has been put into a drive and begins to spin up, the picker can be used to move another cartridge out of another drive.

The transport daemon flushes the asynchronous write operations to the disk before it removes a cartridge from a drive. The spinup daemon flushes all the waiting asynchronous requests to a new cartridge that has just been put into a drive. Thus, all asynchronous writes are eventually flushed out to a drive. If an asynchronous request arrives while a cartridge is currently in a drive, that request is passed to the disk driver immediately. This maintains the asynchronous function of the autochanger driver. One advantage of this daemon approach is that it is portable to other UNIX* architectures.

There are four processes in the autochanger driver: **Accept Requests**, **Schedule Async Requests**, and the transport and spinup daemons. Each process represents a separate thread of execution. **Accept Requests** executes any requests for surfaces already in drives. All other requests are queued. This process is in charge of determining if a swap is to occur as the result of receiving a request by checking the hog time and other conditions. The transport daemon only runs when **Accept Requests** determines that a swap should occur. It also performs the physical unmounting and makes sure the surface in the drive is put in a consistent state before it is removed from the drive. The spinup daemon is charge of spinning up the cartridge after it has been inserted. It also performs the physical mounting and processes any asynchronous requests waiting to use that drive. The **Schedule Async Request** process is executed when no synchronous requests are waiting for a surface. Its purpose is to satisfy the requirement that asynchronous requests return without waiting for any I/O to be performed. This process is not a daemon but is called by an interrupt from the wait timer in the spinup daemon. If the wait timer times out, **Schedule Async Request** is started if no synchronous requests are waiting to use the drive.

*UNIX is a registered trademark of AT&T in the U.S.A. and other countries.

References

1. B. Thompson, D. Stolte, and D. Ellis, "A Transparent Integration Approach for Rewritable Optical Autochangers," *Proceedings of the USENIX Association Summer Conference*, 1990.

Daryl C. Stolte
Bruce A. Thompson
David Ellis
Development Engineers
Greeley Storage Division

(continued from page 10)

payload concept. The ability for the controller to sense forces and distance allowed us to remove physical sensors in the hardware. The architecture called for moving the sensors from the robot back into the information processing capabilities of the motor controller and encoder. Sense of touch algorithms allow the controller to sense whether there is a cartridge in a location, verify a move, and so on.

Another reliability enhancement is overforce sensing and error recovery. Using sense of touch, the robot stops motion before anything is broken. An error recovery algorithm is then invoked to clear the error condition and complete the move sequence. This design provides another chance to avoid a service call.

Details of the servomechanism design are in the article on page 24.

The design is highly modular with limited coupling of the modules. This effort paid off in many ways. The design areas could be assigned to engineers in a relatively uncoupled way, as long as the interactions of the design modules were relatively well-defined. In these design modules, functionality was the key. Designers had the ability to simplify or redesign with measured effect on the rest of the project. This has been very fruitful as the project has continued in its life. As reliability concerns are raised, they can be addressed with minimum impact on other aspects of the design. This allows improvement with limited risk of introducing new problems in related designs.

Trying to achieve minimum system integration time and maximum flexibility led to the concept of mounting the autochanger in either a stand-alone HP rack or a standard 19-inch rack. This would allow the library system to fit into many different systems, both HP and non-HP. This design was adopted early in the project, and is now being born in a follow-on product.

In selecting a system as an initial host, we chose the HP-UX operating system because it provides enough power and flexibility to support this peripheral, and is almost an open architecture. This provided us with a number of options including developing a driver ourselves or developing a file level interface. Integration of the autochanger with the HP-UX operating system is the subject of the box on page 11.

Autochanger Architecture

One of the major goals in the design of the autochanger architecture was defining a growth path for future products. We did not want to lock ourselves into an architecture that would not allow us to meet the interface performance demands of the future. The optical drives that were going to be used in the autochanger communicated via the SCSI, and since there was a standard emerging for an SCSI autochanger command set, we decided that the primary interface of the autochanger would also be SCSI.

There are two drives in the standard configuration, and this means that the autochanger needs to use three SCSI bus IDs (two for the drives and one for the autochanger controller and mechanism). This posed the problem that if more than two autochangers were used by a host on a single SCSI bus, the available bus IDs (8) would soon be used up. We investigated some other architectures that would consume only one SCSI ID, with the autochanger and its two drives configured as logical unit numbers (LUNs) under that single ID. However, there were concerns about the performance degradation of doing the SCSI-to-SCSI command conversion for each LUN.

Even more ambitious than this architecture was the full file-level interface concept. This entailed defining an entirely new interface that interacted on a file level and totally hid the SCSI in the drives. With this concept, we felt we would have complete freedom to optimize the performance, throughput, and thrashing issues associated with an autochanger. On the other hand, this approach would also cause us to abandon the SCSI-II interface standard being adopted by most of the industry, and the use of a standard interface was seen as essential if this product was going to have a viable life as an OEM product.

Although future growth was a major concern, time to market was an even greater preoccupation. The autochanger was on an aggressive schedule, so we needed to be careful about not taking on too big a task for the time allotted. We were able to satisfy both goals by opting for the first architectural option described—the use of three SCSI IDs on the bus.

Autochanger Controller

The autochanger controller board is based on a 68000 microprocessor. The 68000 controls or oversees all processes in the autochanger. Because of the architecture chosen, the 68000 has no direct communication with the magnetooptical drives over the SCSI bus. The autochanger is meant to be an SCSI target device only, and at present does not support any initiator functions.

The 68000 operates with a time-sliced operating system whose primary function is to control the two servo loops of the Y and Z motors. Operation of these two loops can consume up to 70% of the processor's 12-MHz bandwidth. The remainder of the processor's time is involved with command interpretation and the overseeing of all other autochanger functions.

Commands to the autochanger can come from one of three different sources: the SCSI, an RS-232 port, or the front panel. The SCSI is the primary means for controlling the autochanger. The RS-232 port is primarily for diagnostic purposes, although it can also be used as the primary interface for the autochanger controller. The front panel is the personal, direct user interface. Commands can be entered through each of these interfaces, and although their formats differ, they are all processed through a common control flow in the 68000 firmware.

The hardware implementation of this architecture is highly integrated and uses either multifunction or intelligent off-the-shelf parts. The SCSI is managed by a proprietary controller IC, which frees the processor from all but the most necessary processing tasks of the SCSI bus. The RS-232 port is managed by a multifunction peripheral chip. which also handles all the interrupt vectoring and generates various timers used by the controller. The front panel is controlled by an 8051-type microcontroller, which manages all the key presses and is responsible for updating the vacuum fluorescent display. The low-level servo processing is serviced by another proprietary IC, which manages the duty cycle of each motor and monitors the position encoder information.

The controller board also contains 16K words of nonvolatile RAM. This RAM is used to store critical state information and positional parameters for use in autochanger error recovery and powerfail conditions. The nonvolatile RAM also contains certain configuration parameters and certain logging values that constantly reflect the age and health of the machine.

References

1. J.W. Dong, et al, "A Reliable, Autoloading, Streaming Half-Inch Tape Drive," *Hewlett-Packard Journal*, Vol. 39, no. 3, June 1988, pp. 36-42.

Mechanical Design of an Optical Disk Autochanger

The autochanger moves 32 disk cartridges between two magnetooptical drives and two stacks of storage positions using only two motors and three optical sensors.

by Daniel R. Dauner, Raymond C. Sherman, Michael L. Christensen, Jennifer L. Methlie, and Leslie G. Christie, Jr.

THE MECHANICAL DESIGN of the autochanger mechanism for the HP Series 6300 Model 20GB/A rewritable optical disk library system posed several technical challenges, including architecture, reliability, physical size, and schedule. The system holds 32 optical disk cartridges and has two magnetooptical disk drives. The magnetooptical disks are rewritable. Each cartridge holds 650 Mbytes of data; however, only 325 Mbytes is accessible at a time because the drives are single-sided. The total capacity of the library system is 20.8 Gbytes. The system runs on a single-ended SCSI asynchronous bus, which conforms to the SCSI II standard established for autochangers. The average access time to load a disk from a storage position to a drive is seven seconds.

The mechanical architecture of the autochanger excludes any electrical components, cables, or connectors on the moving parts of the mechanism. This "passive payload" concept was chosen to maximize product reliability. The design team set a goal at the onset of the project to have an absolute minimum of sensors, solenoids, and motors. The final design has only two motors, no solenoids, and three optical sensors. The sensors are used in the vertical calibration of the system and in the mailslot.

Physical size was determined early in the product design to allow use in two orientations. In the normal orientation, the autochanger fits into an HP rack. It can also be laid on its side and used in an industry-standard 19-inch rack. This two-orientation requirement established the height, width, and depth of the product. Meeting these space constraints was a persistent challenge in the design of the

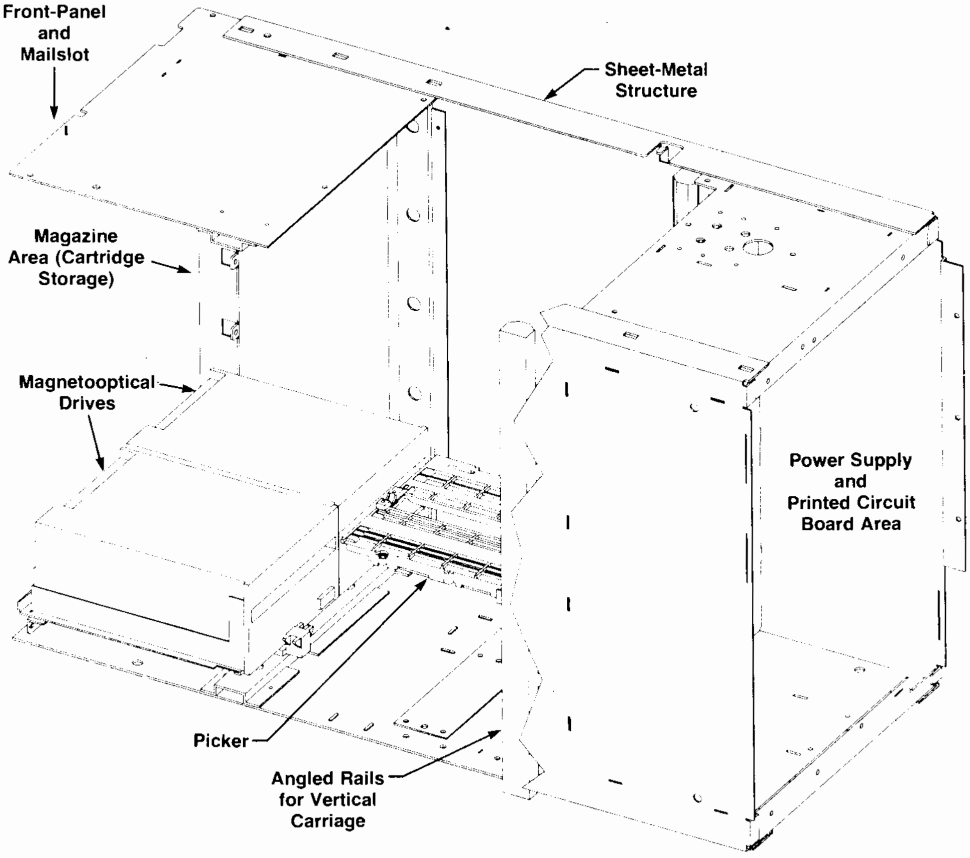


Fig. 1. *Mechanical layout of the HP Series 6300 Model 20GB/A rewritable optical disk library system.*

subsystems.

Adding to these design challenges were the need to ensure HP quality and the time constraints of an aggressive schedule.

Fig. 1 shows the mechanical layout of the autochanger.

Mechanical Functions

There are two basic mechanical functions of the product. First, the user can load or remove a cartridge via the mailslot. Second, the cartridges are moved from storage slots and the mailslot to drives and vice versa via the picker mechanism. These functions are implemented using two motors and associated subsystems. All movement in the product has been grouped into two types: Y or vertical motion and Z or horizontal motion.

Y Motion. All movement up and down along the vertical ways is labeled Y motion. It is driven by a dc servo motor and a vertically mounted leadscrew. A small toothed belt drives the vertical leadscrew through a reduction gear. The vertical carriage is attached to the leadscrew.

The vertical carriage is made up of the horizontal carriage, picker, and translate mechanisms. All of these mechanisms are powered by a toothed belt called the T belt.

Z Motion. The Z or horizontal plane is the plane in which the following motions occur:

- Plunge. The picker plunges to get a cartridge from a drive or slot or to put a cartridge into a drive or slot. The picker is the cartridge carrier, that is, the device that holds a cartridge that is being moved between a storage slot and a drive.
- Flip. The picker is caused to rotate 180 degrees.
- Translate. The system has two stacks of cartridges. In a translate move, the Y and Z systems are positioned to move the picker and the horizontal carriage—on which the picker is mounted—from stack to stack. Translate motions occur only at the lowest vertical position of the vertical carriage.
- Mailslot Actuation. This is a special plunge with picker side and vertical positioning.

All Z motion occurs within or as part of the vertical carriage. All Z motion is driven by the T belt, which is attached to the Z motor and oriented perpendicularly to the vertical carriage assembly.

The plunge occurs in the picker mechanism. This motion is driven by the T belt through a gear attached to the picker leadscrew. The flip is required because the disks are double-sided and the drives are single-sided. The translate occurs through a special combination of Y position and release mechanisms on the vertical carriage. When all of the proper conditions are met the Z motor will drive the horizontal carriage from one stack to the other. Mailslot actuation occurs at a particular vertical height, sensed by mating actuators on the picker mechanism and the mailslot. When all of the proper conditions are met a plunge motion of the picker actuates the mailslot.

Vertical Carriage

As described earlier, the cartridge holder (picker) must be able to move in the vertical direction to any magazine or drive slot. It must also be able to move to one of the two horizontal positions. The vertical carriage is the mechanism that constrains these motions. It is designed to hold the end of the cartridge holder in close tolerance despite variations of the sheet-metal structure that forms the enclosure for the product and to which all of the other parts are mounted and referenced. The vertical carriage is light in weight to reduce dynamic forces. It is designed to be installed and removed easily and to have a high degree of reliability. The biggest challenge in its design proved to be designing it to fit into a 375-mm box structure.

The vertical carriage is shown in Fig. 2. It is guided in its vertical motion by a set of angled rails, which attach to the structure, as shown in Fig. 1. The vertical carriage consists of:

- A set of bearing blocks with roller bearings, which roll on the rails
- Plastic carriage blocks, which hold the bearing blocks, the T belt pulleys, and the horizontal rod and way
- The horizontal rod and way, which provide the translate means for the horizontal carriage and picker.

In the breadboard design, a 0.75-inch-diameter rod and a linear bearing were used for the vertical transport. However, because of the volume limitations, this design proved difficult to implement. To solve this problem and meet manufacturing requirements, the rail and roller bearing approach was chosen. The main problem this design faced

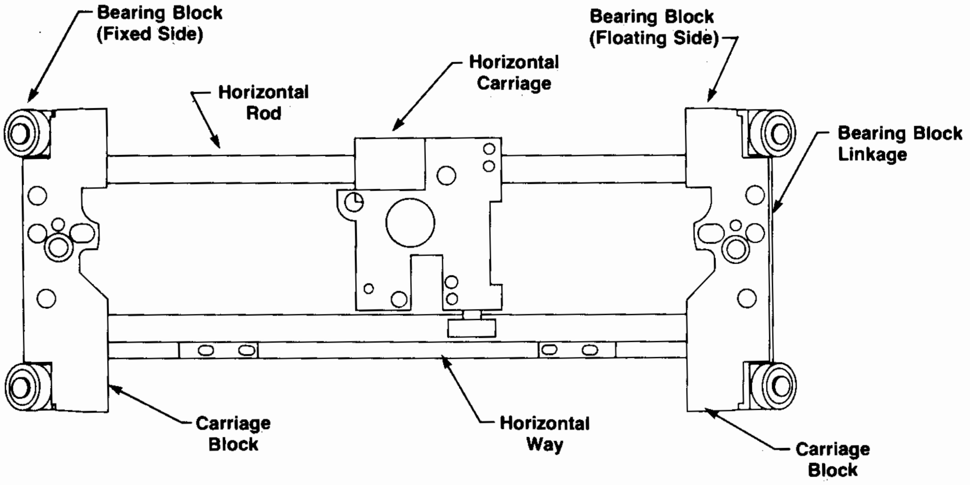


Fig. 2. *Vertical carriage structure.*

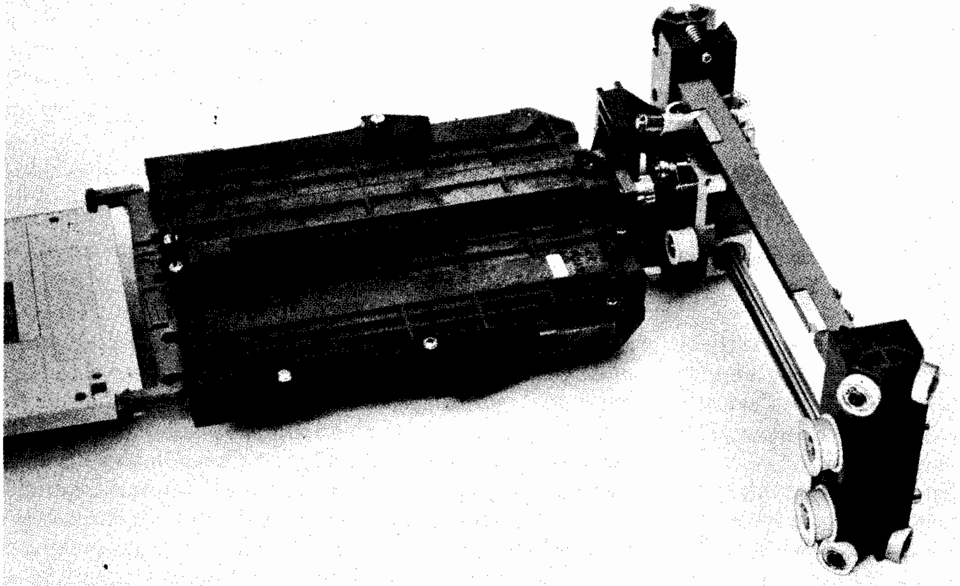


Fig. 3. *Horizontal carriage and picker mounted on the vertical carriage.*

was the variations of the sheet-metal structure. To account for these variations, the right-side bearing blocks are constrained in the carriage block but are spring-loaded (floating) between the vertical rail and the carriage block. This allows the assembly to correct for up to a millimeter of structure variation and still maintain the reference against the left rail, which always serves as the vertical reference surface.

A steel bearing block linkage is used to stiffen the vertical carriage assembly relative to the vertical rails. The bearing block linkage ties the two floating bearing blocks together to ensure that their motion always acts to tighten the vertical carriage within the vertical rails.

Because of the limited volume, the left-side rail has to perform several functions. Half-inch holes in the bottom of the extruded aluminum rail provide for mounting the bearings and shaft. This assembly has space for gearing in the back and the drive gear for the T belt in the front. At the top, a quarter-inch slotted hole in the rail and a plastic slider that fits around it provide a belt tensioner. The top T belt idler pulley is placed through the slider and rail, and is spring-loaded upwards to provide proper belt tension. Because the belt places a moment on the slider, it will lock up when momentary high forces are encountered.

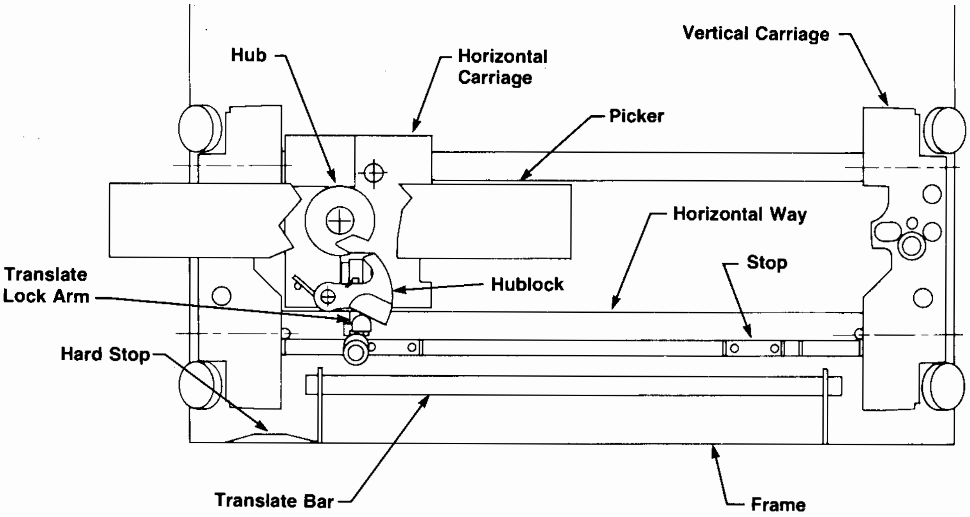


Fig. 4. *Translate mechanism.*

This keeps the belt from slipping when the drive system encounters the high force spikes sometimes seen during magazine and drive insertions.

The bearing blocks provided several design challenges. The first problem was running the bearings on the aluminum rails. This was very noisy and caused the aluminum to wear. Plastic tires were placed on the bearings, but developed flat spots in storage temperature testing. Plastic with better creep qualities was tried, but showed fatigue failures short of the required life.

For the final design, the plastic is Delrin and the tires are redesigned to increase their surface contact area. The bearing blocks were originally designed to be made of aluminum, but the aluminum tended to gall while sliding in the carriage blocks, and was much higher in cost than initially expected. Therefore, an all-plastic design was conceived, using bearing blocks made of polyphenylene sulfide with 40% glass. This design works much better. It results in lower friction, lower wear, better tolerances between mating parts, and a significant cost savings.

The carriage blocks tie the horizontal rod and way together structurally, hold the T belt pulleys, and provide the sliding constraints for the bearing blocks. Glass-filled polycarbonate helps reduce the carriage blocks' weight and cost. Because the carriage blocks are the stops for translate motions of the horizontal carriage, the distance between the blocks is set by machined features in the horizontal rod and way, which are held securely in place by crush bumps in the plastic blocks and then bonded for added rigidity. Once the vertical carriage is installed between the side rails, the carriage blocks cannot separate.

The horizontal rod is a three-eighths-inch hardened stainless-steel rod. The horizontal way at the bottom of the vertical carriage is a machined aluminum L-section bar. The top of the L section is a track for roller bearings on the horizontal carriage and the bottom of the L section provides a latch that holds the horizontal carriage in the appropriate translate positions. The aluminum way also provides a mounting surface for a portion of the translate lock assembly. This way was originally to be extruded, but the added machining made this less cost-effective than a completely machined part.

Horizontal Carriage

The horizontal carriage supports the picker and translates it from one cartridge stack to the other. The support structure for the horizontal carriage allows linear motion in one axis while excluding linear motion in two axes and rotational motion in three axes. A rigid structure is necessary to ensure proper alignment of the picker for cartridge exchanges.

The horizontal carriage is supported by the following parts of the vertical carriage: the horizontal rod, the horizontal way, the two carriage blocks, the four bearing blocks, and the bearing block linkage on the spring-loaded (right) side. Fig. 3 shows the horizontal carriage and picker mounted on the vertical carriage.

A machined aluminum casting controls picker alignment and serves as a mounting surface for the flip support, the translation lock assembly, the hub lock assembly, two idler shafts, two bearings for the main picker shaft, two tire shafts, and two linear bearings for translation capabilities. Rigidity and spatial concerns top the list of design requirements.

The hardened, ground, steel horizontal rod supports the horizontal carriage linear bearings. Two tires on roller bearings, mounted beneath the horizontal carriage, ride on the front and rear surfaces of the aluminum horizontal way (L bar) to control rotation about the rod.

Translate Mechanism

The translate motion—moving a cartridge from one side of the autochanger to the other—requires a combination of the vertical and horizontal motions. There were six design goals for the translate mechanism. First, in line with the passive payload concept, the movement must only use the two servo motors and require no additional sensors or solenoids to clutter a clean and reliable design. Second, the mechanism must be reliable to one million cartridge exchanges. This requires a translate mechanism life of 1.5 million translates because, on the average, there are 1.5 translates per exchange. Third, no more than 6 mm of Y motion can be used to actuate the translate mechanism. Fourth, the design must minimize wear through proper selection of materials and geometry. Fifth, the design must be fault tolerant. This fault tolerance must include the ability to recover from power failures at all times. Sixth, the design must allow translates at both the top and bottom vertical positions to provide architectural flexibility, although in the current product, translates only occur at the bottom.

The translate mechanism (see Fig. 4) consists of the vertical carriage including the horizontal carriage and the horizontal way with adjustable stops, and the translate lock mechanism. The translate lock mechanism (Fig. 5) consists of:

- A translate lock arm, which pivots in and out of a slot in the vertical carriage's horizontal way
- A hublock, which can engage a notch in the picker hub to prevent the hub from rotating
- A translate bar fixed in the structure, which pushes the

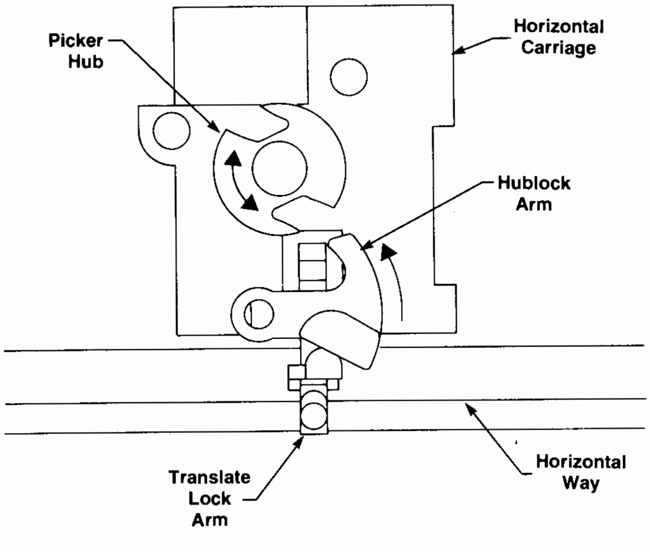


Fig. 5. *Translate lock mechanism.*

lock open.

Both the translate lock arm and the hublock are mounted on the horizontal carriage. They are independently spring-loaded downward for the normal case of the horizontal carriage being locked in place and the plunge motion allowed.

The translate lock mechanism has to hold the horizontal carriage rigidly fixed normally, and ensure that only one motion at a time can occur, either translate or plunge. Ensuring only singular motion is necessary because there are no sensors to tell the system what the picker and horizontal carriage are actually doing, so the servo could conceivably get "lost" without this stipulation. Fig. 6 shows the two basic locked positions: plunge allowed and translate prevented or translate allowed and plunge prevented.

The translate lock arm pivots on a horizontal axis contained within the horizontal carriage. The translate lock arm is spring-loaded to pivot downward, which forces the arm into a slot in the horizontal way and locks the horizontal carriage, and therefore the picker, to one side of the vertical carriage or the other. To minimize wear from the sliding motions, the translate lock arm has a needle bearing on its far end for contacting the translate bar and a molded wear pad closer to the center that contacts the hublock. The translate bar is rigidly attached to the sheet-metal structure and is the contact surface that actuates the translate lock arm. The hublock is another downwardly spring-loaded, pivoting arm that is responsible for locking the hub when actuated by contact with the translate lock arm. The hub is part of the plunge mechanism so that locking the hub results in locking the plunge mechanism. Locking the hub locks the horizontal carriage to the T belt so that the Z servo can move the entire horizontal carriage instead of just actuating the plunge motion.

The translate movement can best be described if split into four phases. In phase 1, the Z motor lines up a slot in the hub with the hublock, allowing for its eventual insertion. The Y servo lowers the entire vertical carriage assembly, causing the translate lock arm to contact and be pivoted upward by the translate bar. The translate lock arm in turn contacts the hublock, moving the hublock into the hub slot and locking the hub and the plunger. At the end of phase 1, both the plunge and translate motions are locked. This overlapping of the two locks is a reliability feature that prevents the Z servo from freewheeling and losing track of where the picker is. The servo is always in positive control.

In phase 2, when the hub is securely locked, the translate lock arm is lifted free and clear of the stops on the horizontal way to allow the Z servo to translate the horizontal carriage across the vertical carriage. Phase 2 is completed when the Y servo saturates* as a result of the vertical carriage's hitting the hard stop of the translate bar at the bottom of the structure, thus ending the vertical movement. No sensor is required to end the vertical movement, thus contributing to reliability and simplicity.

In phase 3, the Y servo is stationary while the Z servo drives the horizontal carriage from one side to the other. The needle bearing on the translate lock arm rides on the translate bar. This allows the tab on the translate lock arm to clear the horizontal way along the entire path. The design has to be tolerant of a possible power failure, since it would be very easy to get into an unrecoverable position at this time. To this end, the translate lock arm geometry is such that if the vertical carriage rises, as it would after a power failure, the Z servo can still pull the horizontal carriage to one side and have the tab on the translate lock arm lock into the stop. This is another major reason why the hublock and the translate lock can never physically be unlocked at the same time, even in the worst-case tolerance conditions. Phase 3 ends with the Z servo saturating with the horizontal carriage against the opposite side of the vertical carriage, thereby finishing the translation part of the move.

The Z servo continues to saturate throughout phase 4, while the Y servo moves the vertical carriage upward. This movement allows the translate lock arm to drop into the beveled stop in the horizontal way, which locks the horizontal carriage. The hublock does not follow the translate lock arm downward because it is held in place through friction by the saturating Z servo. This ensures that the horizontal carriage is in direct contact with the side, allowing reliable locking of the translate lock arm. A bevel in the stop eliminates the possibility that any burr in the lock arm will cause it to hang up. The stop is adjusted during assembly to ensure that the horizontal carriage does not float sideways in the locked position. When the vertical

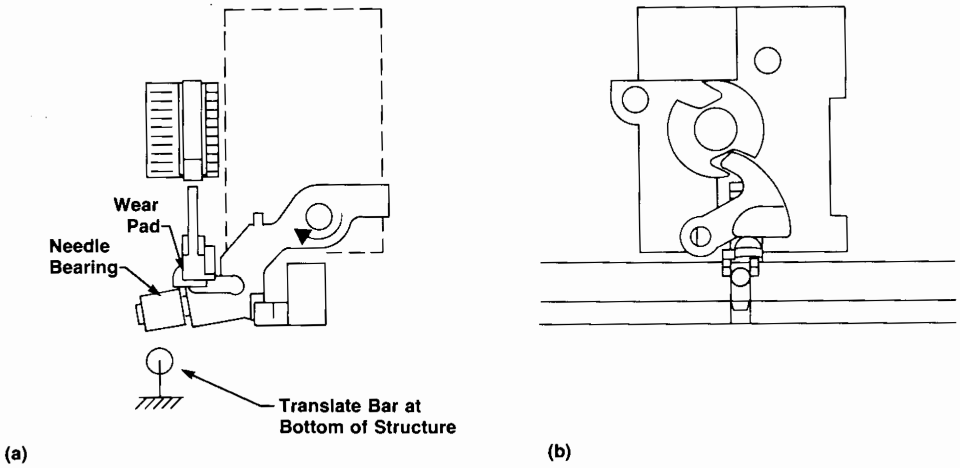


Fig. 6. *(a) Translate lock arm locked into horizontal way. Plunge motions allowed. (b) Hublock arm locked into picker hub. Translate motions allowed—no plunge motions.*

---

*Servo saturation means that the torque output of the servo motor, which is calculated from the motor voltage and the motor torque constant, has exceeded a threshold. Servo saturation is also called force sense of touch.

carriage has moved vertically far enough to allow the needle bearing on the translate lock arm to separate from the translate bar, the Z servo comes out of saturation and allows the hublock to fall back down onto the translate lock arm. The translate move is now complete, freeing the autochanger to perform whatever other moves are requested.

Picker

Like the rest of the autochanger, the cartridge retrieval mechanism is designed with simplicity in mind to help meet both reliability and cost goals. The mechanism requires four degrees of motion within a small form factor: in/out plunge, grasp/release cartridge, side-to-side translate, and 180-degree flip. In spite of all the motions, it was felt that the picker had to have a passive payload, that is, no motors, solenoids, or sensors on the moving platform, for the highest possible reliability.

Fig. 7 shows the basic layout of the mechanism. It consists of six main components or subassemblies:

- The vertical carriage, which connects to the vertical leadscrew, providing the up/down motion
- The horizontal carriage, which holds the cartridge picker mechanism, flip latch, and translate lock arm, and translates from side to side
- The picker hub, which converts the belt motion to picker plunge motion
- The picker itself, which can grab and hold a cartridge
- The translate lock mechanism, which either holds the horizontal carriage fixed or allows it to translate from side to side
- The flip latch mechanism, which holds the picker flat during plunges but can allow the picker to be flipped 180 degrees.

The picker hub is a single plastic molded piece with three functional sections. The back portion is a pulley, which the belt from the motor engages to provide all the power to the horizontal carriage and picker. The front is a gear, which mates with a smaller gear on the end of the picker's leadscrew to convert the belt motion to picker plunge motion. Finally, the center section has two cutouts 180 degrees apart which are used by the translation latch to lock the hub. The belt is the only power source for the entire picker mechanism, and the encoder on the motor moving the belt is the only method of sensing anything on the horizontal carriage.

The picker mechanism design was heavily influenced by space constraints. The width of the sheet-metal structure limited both picker width and picker height (because of flips). Short structure-length constraints and long minimum plunge depth requirements necessitated a compact cartridge grasping mechanism. In addition, the front-panel openings in the magnetooptical drives limited where the picker could grab the cartridge and required that whatever inserted the disk had to go well past the drive front panel. Another concern was how to handle misalignments, particularly as the cartridge was loaded into the drive.

With these concerns in mind, the picker was designed to mimic a hand pushing a cartridge into a drive. "Fingers" grab the cartridge from either the drive or a magazine, and the "thumb" pushes the disk back into place. The leadscrew's nut floats within the thumb, providing the muscle. (The nut is not rigidly attached to the thumb to allow for leadscrew runout and general part variations.) The finger mounts into the thumb, which in turn rides in a plastic shell that has tracks to guide the finger to grab or release the cartridge. Identical thumbs, fingers, and shell halves are used on both sides to form the whole grasping mechanism. Fig. 8 shows the two paths the fingers can take depending on their initial position. The fingers are spring-loaded to a normally rotated-in position. When the fingers

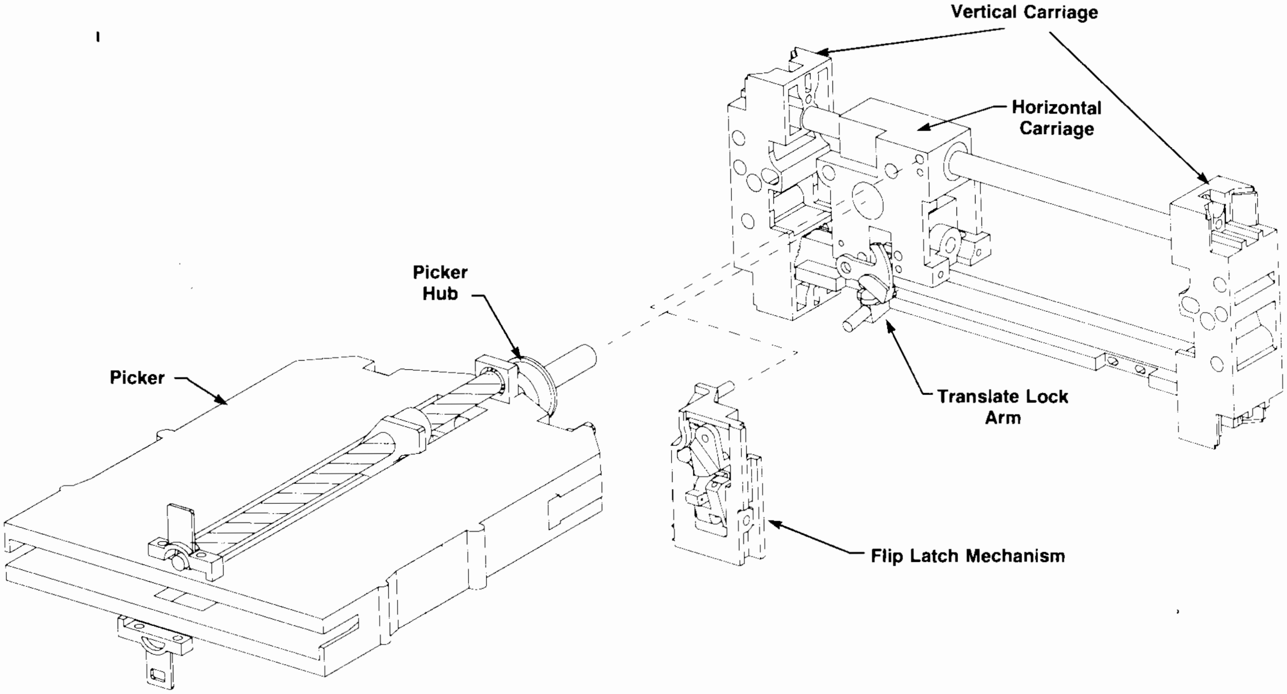


Fig. 7. *Cartridge retrieval mechanism.*

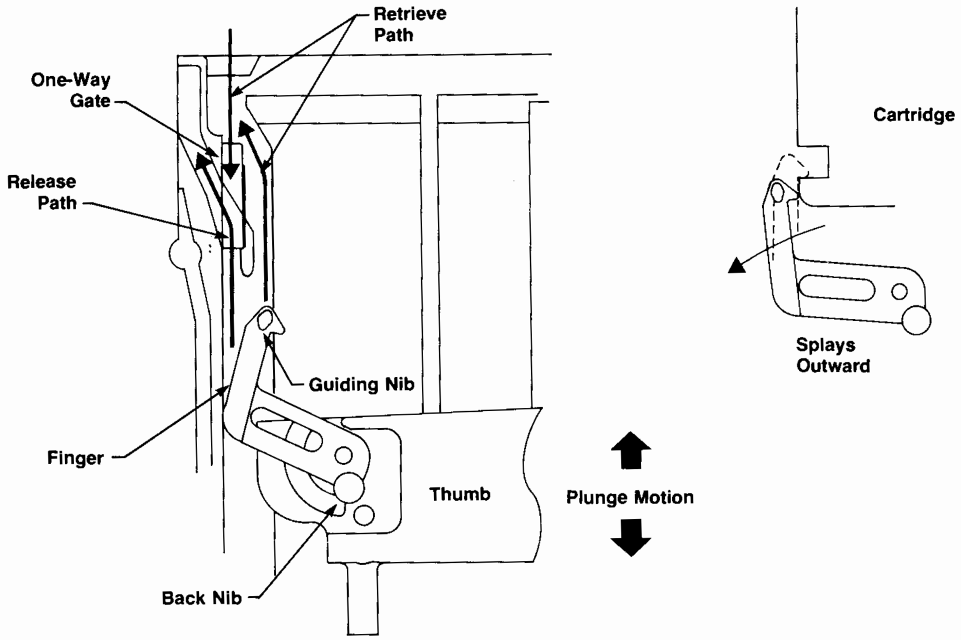


Fig. 8. *Picker finger and thumb mechanism.*

move out and contact the cartridge, they are splayed outward and grab the disk. In the track, a one-way gate turns the system into a type of mechanical flip-flop. This gate, a molded spring, is forced downward as the finger travels over it from the drive side, but will not move down when the fingers come from the picker side. Thus, with the next plunge, the fingers must follow the release path and are rotated out to release the cartridge. The center of the thumb then pushes the cartridge to its final position. As the thumb retracts, the fingers spring around again to their normally closed position, ready to grab a cartridge on the next plunge.

Materials were an interesting challenge in the picker design. The one-way gate is made of Ultem, which has high strength, good wear qualities, and low creep—important characteristics for a preloaded, highly cycled spring. The fingers, thumbs, and sleeves all require low-wearing materials, and since these parts slide against each other, each has to be of a different material. The fingers also require high strength, so they are made of 35% long glass, 15% Teflon polycarbonate. The thumbs have the most parts sliding against them (leadscrew nuts, sleeves, and fingers) and require good flatness, so they are molded of 30% glass, 15% Teflon Nylon 6/10. The sleeves, which are part of the electrostatic discharge path for any charge that might build up on the picker and affect the cartridge, are molded of 10% carbon, 15% Teflon polycarbonate.

Flip Mechanism

The objectives in designing the flip latch mechanism were to use already existing motion and to limit the flip to 180 degrees. Using the rotation of the picker hub without moving the leadscrew satisfies the first objective, and smart hard stops satisfy the second. Fig. 9 shows the main parts of the latch: the pivot arm, the release arm, and the cam. In normal picker plunge motions, a nib on the rear of the picker shell is trapped between the pivot arm on the bottom

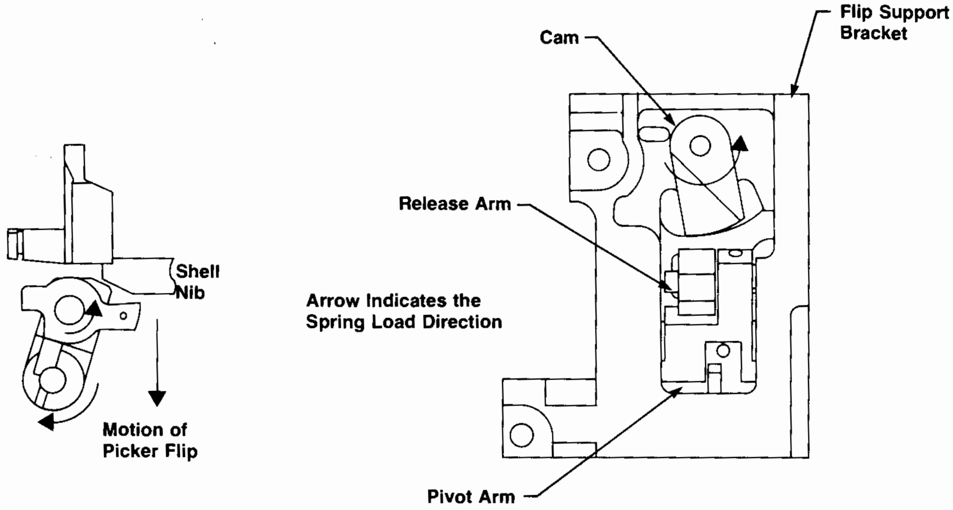


Fig. 9. *Flip latch mechanism.*

and the cam on the top. This keeps the picker flat with respect to the drives and magazines. Fig. 10 shows the steps involved in opening and closing the lock during a flip. To flip, the thumb plunges backwards, pushing the release arm, which is mounted on the pivot arm. Although the release arm can rotate downward on the pivot arm, the thumb, pushing backward on the release arm, forces the pivot arm to rotate backward also, allowing the nib to fall down through. As the belt continues to rotate the hub, the picker's leadscrew bottoms out when the thumb can move back no farther. Thus the entire picker rotates with the hub. As soon as the nib has cleared the pivot arm, the arm springs back to its normal position. At the end of the flip, the nib from the other side comes around, opens the cam, and is stopped by the pivot arm. The thumb, which originally pushed the release arm backward, now pushes it down. With the release arm down, the picker cannot flip again until the lock is rearmed, that is, until the thumb plunges out far enough to allow the release arm to spring back up again.

Stress loads, space, tolerance build-up, and fail-safe once-only actuation were the major concerns in the design of the flip mechanism. Long fiberglass material in both the cam and the pivot arm gives the parts superior wear and impact strength. The orientation of the latch parts not only economizes space but puts the impact loading down onto the pivot shaft for minimal bending stresses. The cam allows for variations in parts and any wear in use while maintaining the picker fixed during plunge operations. The back side of the release arm is designed to help prevent accidental double flips. When the release arm is rotated just slightly, the back no longer aligns with a through hole in the lock's support bracket. This prevents the pivot arm from rotating and eliminates the possibility that the impact at the end of the flip might cause the flip latch to open again.

Mailslot

The mailslot is a mechanism that allows the user to install a cartridge in the autochanger just as if it were being put into a drive. For the picker to grab the disk cartridge, the mailslot mechanism must rotate the cartridge 180 degrees.

The part of the mailslot that accepts and delivers cartridges is called the carrier. In the out position, the carrier extends the cartridge approximately 20 mm out from the front panel for ease of removal by the user. When a cartridge is installed and pushed flush with the front panel, a spring mechanism catches, giving the user a stop position. As this position is reached, a sensor trips, indicating that the mailslot has a cartridge installed. The picker then moves to the correct height to activate the mechanism. Using the picker to activate the mailslot eliminates the need for another motor in the system.

The mailslot rotates the cartridge using forces applied through the picker and an actuator (see Fig. 11). The actuator is approximately at the center of mass of the cartridge and the carrier, thereby keeping the forces in a straight line. The leadscrew nut on the picker drives the actuator. The actuator and the carrier run in tracks molded into the top and bottom pieces of the mailslot. The tracks are designed so that as the actuator drives the the carrier from front to back in the mailslot, the carrier is rotated 180 degrees.

The cartridge load sequence consists of the following moves:

1. User pushes load button
2. Move picker to mailslot actuate height
3. Plunge to maximum get position
4. Check for sensor sensing cartridge in mailslot
5. Rotate mail in
6. Move to get-mail height
7. Plunge to get cartridge
8. Pull cartridge into picker
9. (Sequence of moves to put cartridge into drive or magazine)
10. Move to mailslot actuate height
11. Rotate mail out.

When the carrier is facing the inside of the autochanger, it looks like another magazine slot to the picker. When the carrier is facing the outside it looks like a drive to the user. A special catch mechanism between the actuator and the mailslot top keeps the carrier from moving when either the user or the picker is pushing or pulling on the cartridge, but allows the carrier to be rotated and moved when the picker is pushing or pulling on the actuator. In other words, the carrier has limit positions on both sides of the mailslot, and the actuator must be the moving device to move the carrier past these limit positions.

Because the carrier slides in the top and bottom pieces of the mailslot, the wear of the carrier against these parts required special materials considerations. The carrier and actuator needed to be different from the top and bottom, but all parts were to be molded. For regulatory reasons, all

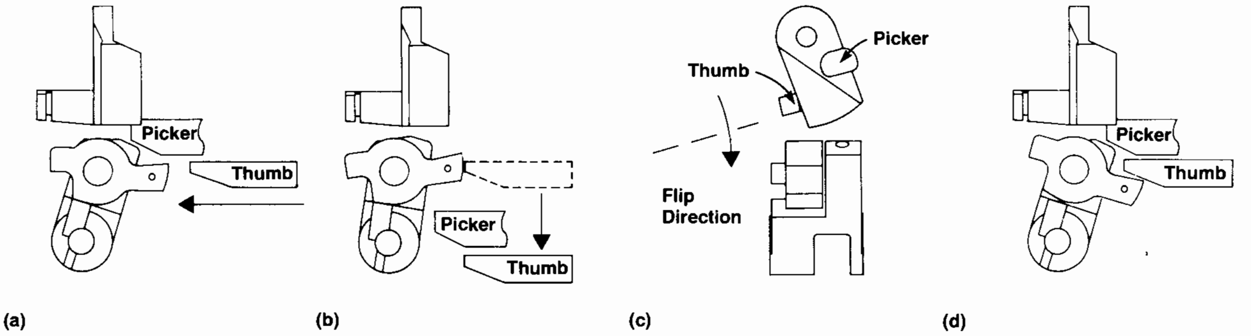


Fig. 10. *Flip latch operation. (a) Initiation of flip. Thumb pushes release arm. (b) Start of flip. Picker falls through. (c) Start of cam opening. (d) End position of flip. Release arm cocked.*

of the materials had to be self-extinguishing when exposed to flame. The top and bottom are made of polycarbonate with glass and Teflon fillers. Polycarbonate is used because of its low cost, since these are the two largest parts. The Teflon (PTFE) is put in for friction reduction. The glass is added because the parts need to be flat and strong enough to hold the whole assembly while the user or picker pushes on the mechanism from either side. A special milled glass fiber was chosen as a filler because it tends to produce flatter parts (±0.2 mm across the part) and adds the required stiffness. The carrier and actuator are both molded out of polyethersulfone (PES) with Teflon filler. PES has several characteristics that are needed in the part design. It is very dimensionally stable material. The PES and polycarbonate materials wear against each other very well. PES can be color matched to the custom color required by HP. Also, PES can be ultrasonically welded. The carrier is approximately 120 mm deep over a section 11 mm high. Molding this would be very difficult if not impossible, so the carrier is made in two pieces and the two are welded together.

To assemble the mailslot, all parts are either added to the top half of the enclosure or placed directly into the bottom half. The top and bottom are then screwed together.

Magazines

The four magazines in the HP Series 6300 Model 20GB/A rewritable optical disk library system each hold eight magnetooptical disk cartridges (Fig. 12). The magazines must hold the cartridges in position for the picker to remove and replace them without too much force, but they must also hold the cartridges while the machine is physically moved. The magazines are designed so that referencing and alignment are correct when the assembly is inserted into the structure.

The alternative of molding the magazine assemblies in one part was ruled out because of schedule constraints and tooling costs. Minimum wear on the cartridge contact surfaces required a plastic part, and for ease of assembly we use a sheet-metal mating part. The entire magazine assembly uses only three parts: a plastic part used four times, a sheet metal part used twice, and a plastite screw used eight times. Details needed for holding the cartridges are easily created in the plastic part. Sheet metal is an inexpensive and effective way of holding the plastic parts together and referencing them into the structure.

The plastic guide in the magazine assembly is made of polycarbonate with 10% PTFE (Teflon) and 10% aramid (Kevlar) fibers. The Teflon is added for friction reduction. Testing during product development showed that the standard cartridge case made of ABS wore too easily against even the lubricated guide, so the cartridge cases were changed to polycarbonate. Extensive testing has shown making both the cartridge case and the guide of polycarbonate is acceptable because of the lubrication in the guide.

The aramid is an additive for dimensional stability. During molding, the aramid ensures that the part shrinkage is the same in both flow directions. This keeps the plastic guide stable in all of its required referencing tasks. The cartridge is spaced and held vertically by the guide. The catch details for holding the cartridge are molded into this part and close tolerances are needed for a proper snap fit. The plastic molding process allows a design that lowers insertion force but keeps the removal force at desired levels. Fig. 12 shows the cantilever spring and cartridge snap details.

The autochanger is designed for a million exchanges.

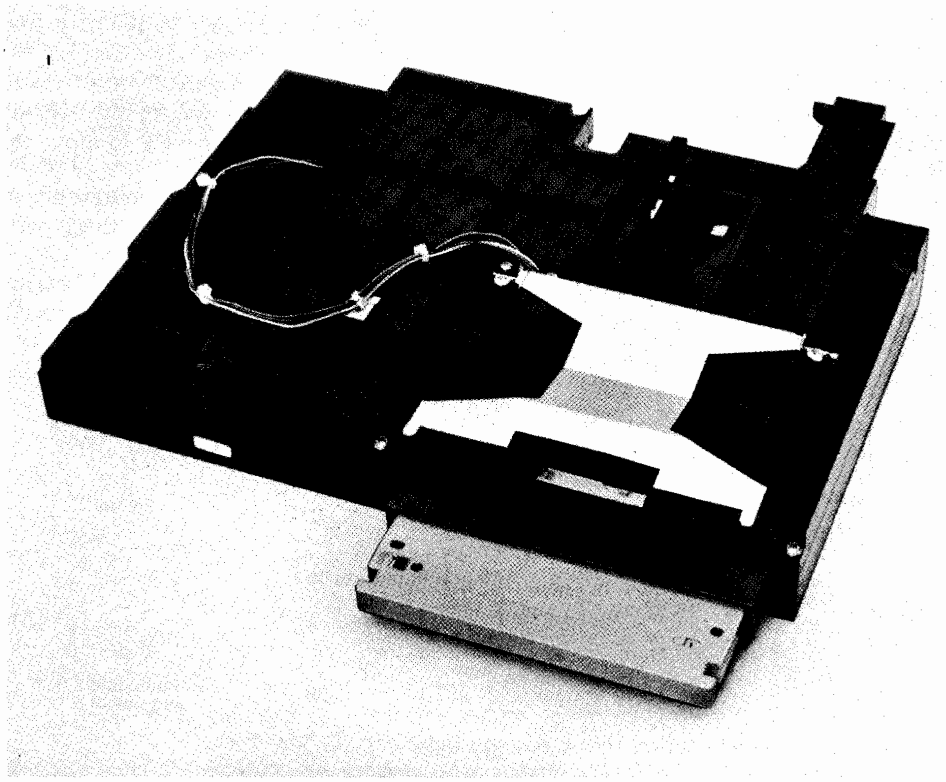


Fig. 11. *Mailslot assembly.*

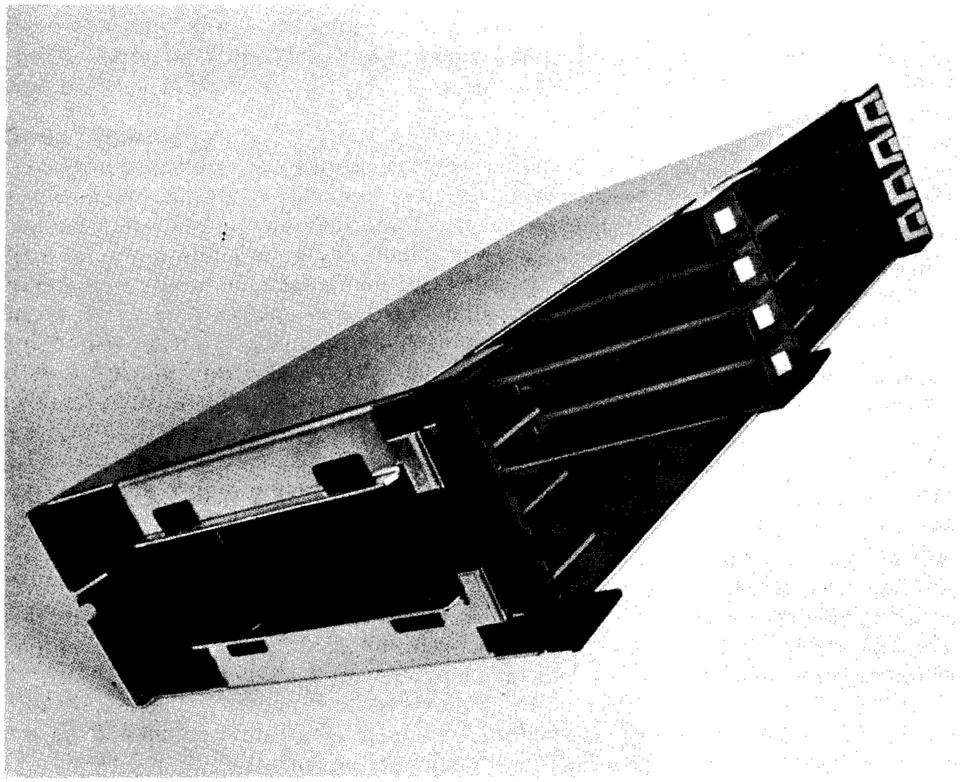


Fig. 12. *One of the four eight-cartridge magazines. Cantilever springs and snap details to hold the cartridges are molded into the plastic side parts of the magazines.*

Therefore, the springs designed into the guides must have a long fatigue life. The Kevlar is helpful in this area, but not as much as fiberglass would be. A trade-off of dimensional stability for part strength and therefore fatigue life was made in the choice of materials. This trade-off has required testing to ensure that the spring life is at least 150,000 cycles per spring. Considerable testing has established that the part meets the requirement. The cantilever springs are designed for constant stress over their entire length.

The sheet-metal part is a horizontally symmetric part that holds the plastic parts together and allows easy installation of the assembly into the structure. This part is folded such that the side plastic parts are accurately located with respect to the reference surfaces on the sides of the structure. This reference scheme keeps the tolerances between the magazines and other critical elements to a minimum. Once the magazine assembly is built, it can only be installed correctly into the machine. Hard-tooling of the sheet-metal part reduced magazine location errors and improved performance of the unit during of the course of testing and design.

Acknowledgments

Special thanks to Dave Jones and Doug Fleece, whose efforts helped this project meet its goals. Dave, who worked on the project from the beginning, designed the sheet-metal enclosure and did the product design. He was also a great help with a lot of good ideas in many areas. Doug, who came on the project later and finished up in some needed areas, designed the front bezel and mailslot door and finished up several smaller parts.

Optical Disk Autochanger Servomechanism Design

A "sense of touch" and error recovery routines contribute to reliability. Data capture, error injection, and mechanical regression testing facilities improved the productivity of the designers.

by Thomas C. Oliver and Mark J. Bianchi

THE SERVOMECHANISM OF THE HP Series 6300 Model 20GB/A rewritable optical disk library system is a collection of electronics and firmware algorithms that control the autochanger mechanism. The servo provides the muscles and brains that bring the mechanical limbs to life. Muscles are provided using motors, power supplies, and sensors. The brains are contained in the firmware program that controls how the muscles are energized.

It is the responsibility of the servo to control the autochanger mechanism reliably. Designing for reliable control requires that many aspects of the system be analyzed and optimized. The design encompasses a broad range of engineering disciplines, including system models for stability and performance, continuous and discrete-time control theory, firmware architecture, motor parameter optimization, analog hardware design, and digital logic design.

Goals and Solutions

The goals for the servomechanism design were high reliability, flexibility, system stability, high performance, user safety, and contributions to other parts of the development effort.

Techniques for achieving high reliability include the sense of touch for adaptive, gentle movements, minimizing the number of components through firmware integration, the use of proven, reliable technologies (HP ICs, standard LSI, surface mount technology), increasing hardware design margins by overrating, and the use of self-calibration, error detection, and recovery techniques.

Flexibility is provided by firmware implementation of servo functions, a modular design architecture, and the use of a high-level programming language.

System stability is ensured by extensive system modeling before and during implementation, optimized programmable compensation for each movement, and margin verification over all operating ranges.

High performance is achieved by optimal compensator selection for each movement, motor and power supply optimization based on the performance model, and overlapping of movements.

User safety is ensured by continuously monitoring applied forces in firmware and hardware.

Contributions to the development project outside the

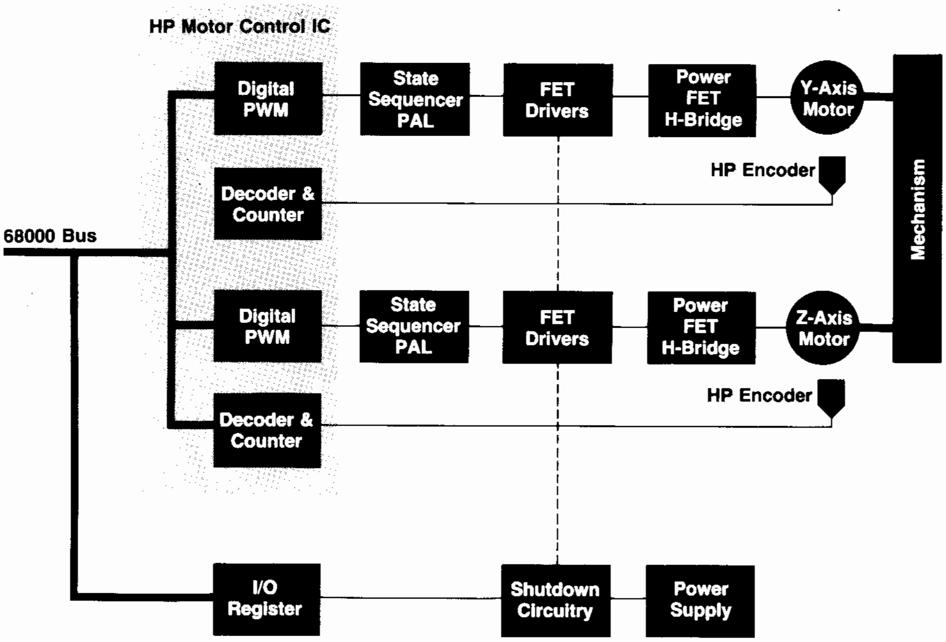


Fig. 1. *Servo hardware architecture of the HP Series 6300 Model 20GB/A rewritable optical disk library system.*

servo area include a data capture system, mechanical regression tests, error injection, and code that can be leveraged by other projects.

Design Philosophy

Extensive modeling was performed before the design was implemented. A performance model was developed so that motors, gears, and power supplies could be selected to meet the swap time goal. Plant models and compensator techniques* were simulated so that optimal control schemes could be investigated. Bandwidth analysis was required to ensure that a microprocessor could close the control loop fast enough. Root locus and Bode analysis methods were vital tools used to ensure adequate stability margins.

Proven, reliable HP technologies are employed in the servo design. Digital implementations were selected over analog techniques because they offer greater flexibility and fewer components. A custom digital IC from the HP DraftPro plotter family is used because it has a proven track record and is used in large volumes. HP optical encoders were a natural choice based on their exceptional reliability and manufacturability. A power driver from the HP 7980A tape drive is also used. Design margins were increased on all critical components, particularly the devices that dissipate large amounts of power. Additional reliability is achieved by sharing a single microprocessor between the servo and interface functions.

The servo firmware architecture contributes to HP Series 6300 Model 20GB/A reliability on many levels. The firmware is designed to provide maximum integration of servo functions such as closed-loop control, profile generation, and error detection. Firmware integration helps reduce parts count and increase flexibility. A "sense of touch" technique was developed to eliminate the need for sensors on the moving transport, a key contribution to reliability. Control of each mechanical function (vertical movement, flip, translate, I/O) is tailored to provide gentle, adaptive movements. Self-calibration, error detection, and error recovery play key roles in increasing reliability and manufacturability.

Another project philosophy was to increase the productivity of the design team through the creation and use of tools. Features such as data capture, error injection, and mechanical regression tests are incorporated into the firmware to give the design engineers a "mechanical oscilloscope" for the mechanism. Our investment in networked HP 9000 workstations provided a common development, debug, and testing platform for the design team.

Hardware Architecture

The servo hardware is kept to a minimum to ensure reliability. Digital circuitry is used whenever possible because it usually requires fewer parts and is more flexible than analog implementations. Real-time functions, which can't be performed in firmware, reside in hardware. These functions include the motor drivers and the motor position encoding. Fig. 1 shows the servo hardware architecture.

The motor driver consists of a pulse width modulator and an H-bridge amplifier. A custom HP ASIC (application-specific IC) is used to generate a PWM (pulse width modulation) signal for two motor drivers configured for bidirectional operation. The IC contains a register that is used to transform a digital value into a TTL PWM signal having a duty cycle proportional to the register value. A state sequencer PAL (programmable array logic) transforms the IC's outputs into four time-sequenced signals that control operation of the FET H-bridge. The PAL prevents cross conduction in the FET H-bridge and offers a flexible alternative to analog time delay circuits. The FETs amplify the sequencer outputs and present voltage pulses to the motor. The motor averages out the high-frequency pulses and responds as if a dc voltage were applied at its inputs.

As the motor turns, its two-channel shaft encoder sends a series of pulses to the HP ASIC. Quadrature decoding is performed by the ASIC, and a register representing the motor position is made available to the system. Thus, the firmware uses the ASIC as a single interface with which

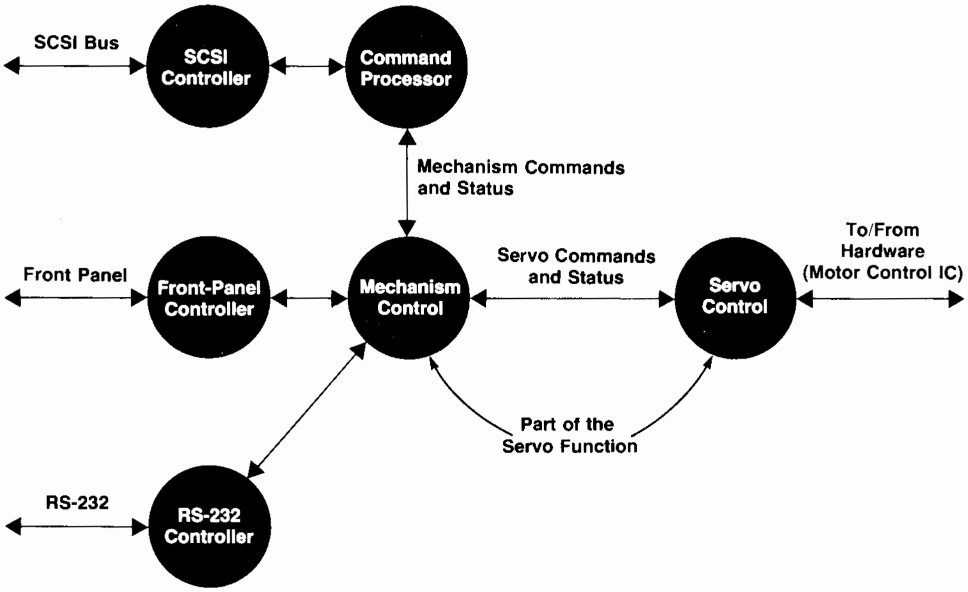


Fig. 2. *Top-level firmware architecture.*

---

*A plant model is a control theory model of the device being controlled. Compensator techniques are methods of stabilizing the control system.

to control the motor voltage and receive positional feedback.

Firmware Architecture

The servo firmware is partitioned into two types of code: real-time and nonreal-time. Real-time firmware is referred to as servo code while nonreal-time firmware is called mechanism code. Servo code is responsible for the closed-loop operation of the system. Real-time code is placed in an interrupt routine, which executes once every millisecond. Mechanism operations are coordinated by the mechanism code through control of the servo code. The two processes communicate through a set of primitive routines, which manage the flow of commands and state information. Fig. 2 shows the overall firmware architecture and Fig. 3 shows the servo firmware architecture.

Servo Code

The servo code is responsible for closed-loop compensation, profile generation, sense of touch calculation, error detection and shutdown, and development tools.

Closed-Loop Compensation. Closed-loop compensation is the algorithm that determines how the motors respond to command position changes or payload disturbances. Compensation is implemented with a simple P-D (proportional-derivative) algorithm. It can be described by the following equation:

$$\mathrm{pwm} = K_1 (e - K_2 \omega),$$

where pwm is the command voltage to the PWM on the motor control ASIC, e is the motor's position error, ω is the motor's angular velocity, K_1 is the proportional gain (stiffness), and K_2 is the derivative gain.

Since each mechanical function has a different load characteristic, the servo gains must be adjusted for each mechanical operation. Knowledge of the load is maintained by the mechanism code, which controls these gain values.

Profile Generation. The method used to change the motors' position from point (Y_1,Z_1) to point (Y_2,Z_2) is referred to as profile generation. Profile generation is the creation of a series of command positions to the Y and Z compensators that will cause the motors to move a desired distance. A trapezoidal velocity profile is generated based on parameters of distance, acceleration, and peak velocity. Scaling is also performed to allow the motors to move different distances at different speeds simultaneously.

Sense of Touch Calculation. The servo code is responsible for determining the forces being exerted by each motor/load system. Forces are monitored by the servo and mechanism firmware to obtain an additional form of feedback from the mechanism. During each interrupt, the servo calculates force using an equation based on motor voltage and velocity. The equation is:

$$f = K_3 \mathrm{pwm} - K_4 \omega,$$

where f is the applied force, pwm is the command voltage to the PWM on the motor control IC, ω is the motor speed, and K_3 and K_4 are constants based on the motors and gearing. Traditional techniques use direct current measure-

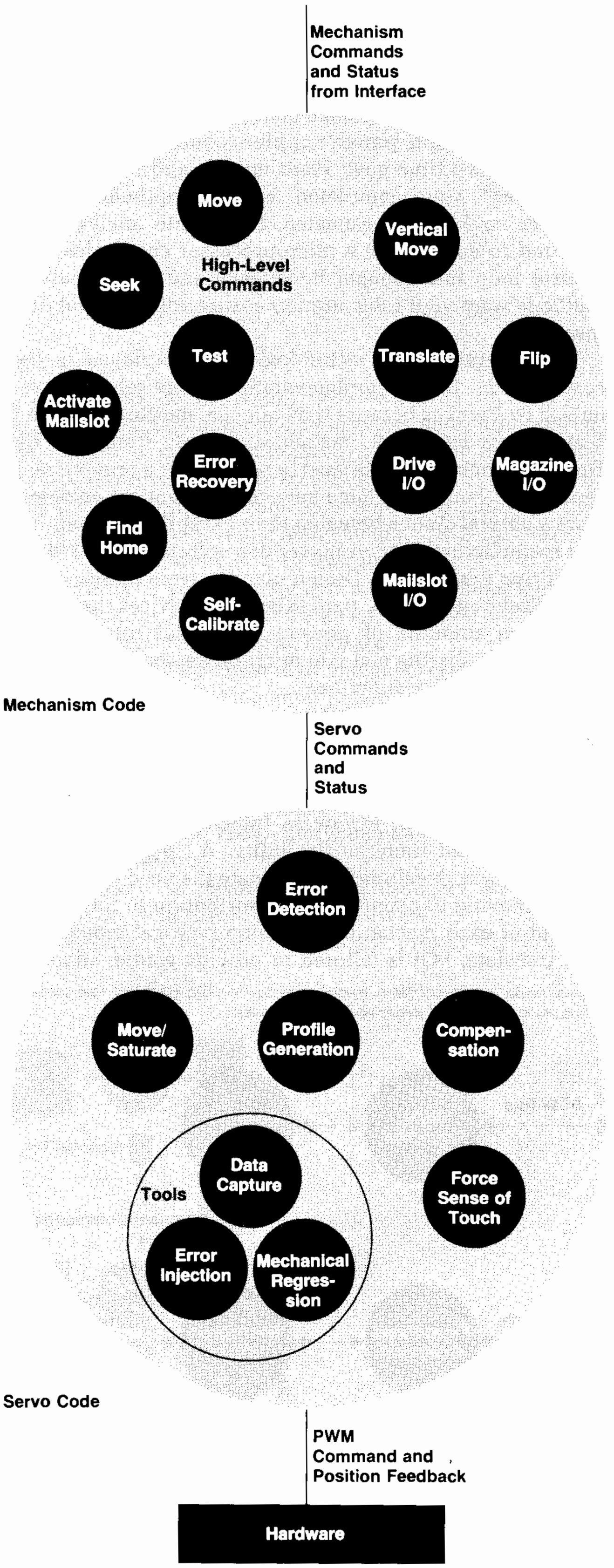


Fig. 3. *Servo firmware architecture.*

ment, which involves added circuitry and cost.

Error Detection and Shutdown. Error detection is an important safety and reliability feature of the servo code. During each interrupt, the servo firmware monitors the forces, voltages, and currents of both motor systems to determine if an error condition exists. Measured values are compared against expected thresholds for a given submove. The mechanism control code tailors the threshold for each submove within a mechanical operation. If limits are exceeded, the servo will immediately disable motor power and record pertinent information for later processing.

Development Tools. The servo code also performs functions that aided in the project as a whole. Features such as data capture, error injection, and peak force detection were designed into the firmware in its early stages. Data capture provided designers with a multichannel "mechanical oscilloscope" to observe key variables used in the servo interrupt (see "Data Capture," page 29). Error injection allowed designers to simulate error conditions within the servo code and observe the system's response (see "Error Injection," page 33). Peak force detection provides servo information that was used to identify movements that were stressing the design margins of the mechanical assemblies. Hooks for the mechanical regression test were used to determine how well a unit was operating over time.

Mechanism Code

The mechanism code provides a translation between the SCSI interface and the servo code. Commands are received and executed by transforming them into a series of smaller submoves. Adaption, error detection, and recovery functionalities also reside in the mechanism code.

The mechanism code accepts high-level commands from the autochanger interface, executes the command, and returns status. High-level commands include the following:

- Move/Exchange. Move cartridge from element A to element B.
- Seek. Position transport at target element.
- Test. Test for the presence of a cartridge at a target element.
- Actuate Mailslot. Rotate the mailslot assembly to perform I/O with the user.

An element is defined to be any possible resting place for a cartridge, including storage magazines, optical drives, the mailslot, and the transport.

The commands are transformed into a series of basic autochanger operations, such as:

- Vertical Move. Position transport to a vertical position.
- Translate. Position transport to access a given vertical stack of cartridges.
- Flip. Rotate the transport.
- Cartridge I/O. Control the plunger to move cartridges between the transport and the magazines, drives, or mailslot.
- Rotate Mailslot. Control the plunger to rotate the mailslot assembly to or from the user.

For example, "Move element 11 to element 2 with flip" would be transformed into the following sequence of autochanger functions:

1) Determine that element 11 is a storage slot and element 2 is a drive.
2) Determine if a translate is required to position the transport to the appropriate stack for the storage slot. If so, perform the translate.
3) Perform vertical move to the storage element.
4) Get cartridge from the storage element.
5) Perform flip.
6) Determine if a translate is required to position the transport to the appropriate stack for the drive. If so, perform the translate.
7) Perform vertical move to the drive element.
8) Put cartridge into the drive element.

Each autochanger function is then transformed into a series of small movements called submoves. There are two types of submoves:

- Move (Y,Z). Move motors a given distance at a specified acceleration and peak speed.
- Saturate (Y,Z). Same as a move except that motion is halted if force exceeds a specified threshold.

Move (Y,Z) is used for high-speed, unobstructed movements of a known distance. Saturate (Y,Z) is for low-speed, adaptive movements of variable distance.

Autochanger functions consist of one or more combinations of move (Y,Z) and saturate (Y,Z) submoves. Each function has a tailored set of these submoves that guarantees that the movements will be gentle. As the submoves are executed, the servo gains are adjusted to allow for changes in load characteristics. An example of the process for a flip is outlined below:

1) Move plunger backwards a fixed distance to engage the flip lock.
2) Change gain for flipping.
3) Move plunger backwards a fixed distance to perform the flip.
4) Ensure that the flip is completed by performing a saturate until the force exceeds a fixed threshold.
5) Change gain for plunger movement.
6) Move plunger forward to relieve the force.

Each submove within a function has a unique set of stability, performance, error recovery, force, and reliability critera. Each submove is assigned a unique identification code (ID) which is used to determine how the move should be performed. Before a submove is executed, its ID is used to fetch acceleration, velocity, and force limits to use. If the move fails, its ID is used to determine the type of error recovery scheme to employ. This tailored technique provides gentle, stable control of the mechanism, thus increasing reliability.

Adaption

Adaptive algorithms were developed to increase reliability and decrease sensitivities to dimensional variations. Dimensions that require adaption are the translate distance, flip distance, magazine depths, mailslot depth, and mailslot actuator throw. If a dimension is susceptible to variation, the firmware is designed to measure the distance using a two-step process. The first step is to undershoot the typical position using a move. The second step is to perform a saturate until a hard stop is encountered. The amount of variation is calculated and the proper dimensional adjustment is made. Subsequent operations will be performed at full speed using the newly calibrated dimension. This form of self-correction eliminates unnecessary impulse forces caused by tolerance buildup.

Adaption is also used to increase the reliability of the autochanger/drive interface. Initially, drive insertions are performed at slow speeds. An adaptive technique is employed to measure the point at which the drive accepts the cartridge. The results are used so that subsequent insertions are exact and can occur at high speed.

Mechanism Initialization

For the servo firmware to be able to perform controlled movements, the mechanism must be methodically set into a known initial state. This process of initializing the transport system is referred to as finding home. The successful completion of the find home routines is crucial for the proper operation of the autochanger.

The transport mechanism is designed to operate through a number of degrees of freedom using only two servo motors. As a result, motions such as flip, translate, and insert/extract require the servo to move the transport thumb to specific absolute positions within the transport sleeve (see article, page 14). The translate motion, in addition to requiring a specific location along the transport sleeve, requires an absolute reference point at the bottom of the autochanger. These positional requirements necessitate the accurate and repeatable location of two points of origin from which the servo can reference its motions. These are called the plunge origin and the vertical origin. Once these points of origin are found, the servo can reliably perform all of its fundamental movements. However, the autochanger may have power removed at any time during one of its motions. This implies that upon subsequent restoration of power, the mechanism may not be in a state that facilitates locating these points of origin. The find home algorithms must therefore be capable of interpreting the current state of the mechanics through various feedback methods, moving the mechanics in such a way that location of the points of origin is possible, and finally locating the points of origin in a very repeatable and reliable manner.

To interpret the current state of the mechanics upon power-up, the find home process employs a number of algorithms collectively referred to as initial recovery. These algorithms are charged with assessing the state of the mechanics using position feedback and force sense of touch. Once sufficient information has been gathered, these algorithms are also responsible for maneuvering the mechanism to a position from which it is possible to complete the find home process. Each initial recovery algorithm is designed to perform specific motions assuming a certain mechanical configuration and/or range of position. Therefore, each algorithm is most effective when used with a specific move type. To choose the recovery algorithm that best matches the current state of the autochanger, the nonvolatile RAM is examined to determine the ID of the last movement that was occurring when the power was removed. This number is used to select the appropriate initial recovery routine.

Once the initial recovery routine has completed, the find home process can proceed with locating the plunge and vertical origins. This process involves a number of steps that must be performed in a specific order. The transport must complete a full flip so that the mechanical hard stop along the plunge axis can be located. Force sense of touch is employed to locate this point and the servo firmware initializes the plunge axis position to zero. The transport can then be moved downward to locate the vertical hard stop at the bottom of the autochanger. After this is determined using force sense of touch, the vertical position is initialized to zero. The last basic find home motion is performed by sensing which vertical stack the transport is facing. Once the two axis origins have been located and the correct stack has been set, the mechanism is able to perform all of its fundamental motions.

Three more operations are performed that provide the firmware with additional information. The first involves a slow traversal of the two vertical stacks while monitoring the vertical force. This motion ensures that the entire path is free of obstructions. The second involves determining whether there is a cartridge in the transport sleeve. This is performed by moving the plunger slowly outward towards a solid section of the mailslot while monitoring the force exerted. The presence or absence of a large force increase denotes the presence or absence of a cartridge. The third operation involves determining which side of the transport sleeve is facing upward. This piece of information is important since it helps determine the orientation of the cartridge before it is inserted into a magnetooptical drive. These drives are single-sided devices, so proper orientation of the cartridge is vital to the successful completion of a move or exchange command.

The accurate positioning of the front of the transport is critical for reliable insertion and retraction of cartridges into and from the magazine slots and magnetooptical drives. The servo system is capable of very accurate positioning of the transport mechanism. However, vertical motion of the transport is controlled from the horizontal carriage assembly, which is in the rear of the transport. Mechanical tolerances and variations in the manufacturing of the transport assembly may result in the mechanism's not being exactly perpendicular with the vertical axis. In addition to a deviation from perpendicular, the front of the transport may change its vertical position in flipping from one side to another and from changing from one vertical stack to another. To compensate for these three variations, two optical sensors are employed in conjunction with firmware algorithms to calibrate the transport system.

Transport Calibration

The accurate positioning of the front of the transport is critical for reliable insertion and retraction of disks into and from the magazine slots and optical drives. The servo system is capable of very accurate positioning of the transport mechanism. However, vertical motion of the transport is controlled from the horizontal carriage assembly, which is in the rear of the transport. Mechanical tolerances and variations in the manufacturing of the transport assembly result in a mechanism that may not be exactly perpendicular to the vertical axis. In addition to a deviation from perpendicular, the front of the transport can change its vertical position in flipping from one side to another and in changing from one vertical stack to another. To compensate for these three variations, two optical sensors are em-

Data Capture System

Early in the development of the autochanger, it was found that much of the servomechanical testing and evaluation could not be performed by visual observation. The unaided eye was sufficient to diagnose gross mechanical problems at slow speeds. However, the design team needed some way to instrument the autochanger so that servo and mechanical parameters could be accurately correlated and analyzed. Typical methods of measurement would have involved the use of accelerometers and/or high-speed cameras to provide dynamic measurements. These techniques were dismissed since they could not provide many of the measurements needed by the team, and because of the difficulty of synchronizing their output with simultaneous measurements of the servo loops. A "mechanical oscilloscope" was deemed necessary to facilitate the debugging of high-speed, dynamic problems and to assist the designers in the evaluation of design modifications. Thus, the data capture system was born.

The data capture system is a combination of firmware-resident procedures and workstation-based tools. It provides the designers with the means to examine the variation of any important firmware variable with respect to time. The capture system employs the HP 64000 emulation system in conjunction with "homegrown" data processing and plotting tools. It is designed to be used in a windowed environment, since its output is displayed as an X-Y graph of the variables of interest. Autoplot, a very flexible, general-purpose plotting program, displays the output data in a graphics window for quick viewing. The beauty of the system is that no special hardware is needed, assuming that one has a workstation and an emulation system. In our case, each firmware designer had both of these, so each designer also had a complete data capture system. This greatly improved the team's productivity and ability to debug complex, dynamic servomechanical problems.

Since the servo system contains accurate position information and is operated on a repeatable time base (one-millisecond interrupt cycle), it is an excellent choice for a mechanical measurement system. Positions, velocities, accelerations, and forces are accurately measured in the servo loops and are maintained in digital format. The data capture firmware exploits this by simply copying the values of these variables into a buffer during the servo interrupt cycle. In this way, a log of sampled data is created that can be processed into graphical format.

The data capture system allows a number of variables (up to 10) to be traced simultaneously during a user-specified length of time. The sampling period can be set from the highest resolution of one sample per millisecond down to the lowest of one sample per 255 milliseconds. In addition, the trigger event that begins the data trace can be set to one of four different conditions: (1) the first occurrence of a specified move ID, (2) the first occurrence of a specified error code, (3) the occurrence of any error condition, or (4) a special event that can be inserted into the firmware specifically for debugging. The trigger location within the capture time is also user-definable and can be set at the beginning, the center, or the end of the capture period. This feature makes it possible to view data that immediately follows the trigger event, data that surrounds the trigger event, or data that precedes the trigger event.

Data Capture Operation

Fig. 1 shows how the data capture system operates. The capture parameters for a specific trace (variables of interest, sample period, capture time, and trigger condition and position) are entered into a text file. A shell script is invoked that converts this text file into an HP 64000 command script. After executing this command script in the emulation window, the data capture system is armed and awaits the trigger condition. At this point, the data capture routines begin copying the values of the specified variables into the data capture buffer, which is located in an unused area of emulator RAM. These routines copy the variables into sequential memory locations in the buffer every sample period. The size of the capture buffer is determined by the number of variables, the size of each variable (byte, word, or long word), the time between samples, and the capture duration. The capture firmware uses pointer arithmetic to keep track of the beginning and end of the buffer, and to implement a circular buffer. Once the capture system is enabled, the autochanger is instructed to perform the motions that will cause the trigger event to occur. When the trigger condition occurs, the remainder of the capture buffer is filled with data.

At this point, another command script is executed in the emulation window. This script copies the contents of the buffer memory to a file. This data file is then processed by a program called mkplt, which generates autoplot commands as its output. The information displayed in the output plot is determined by parameters located within the data capture configuration file. These parameters define which captured variables should be plotted on each axis, what scaling factor should be applied to each variable, and what titles should be placed on each axis. Mkplt also allows simple math functions to be performed on the variables (e.g., plot "var1 – var2" or plot "– var3"). Any of the captured variables can be used as the independent variable instead of time. For example, it is possible to plot the measured vertical force along the Y axis and the measured vertical position along the X axis. The mechanical regression utilities make use of this feature to generate a plot of friction versus position along a given mechanical axis.

A new data trace can be created by reexecuting the HP 64000 command script, exercising the autochanger, saving the buffer

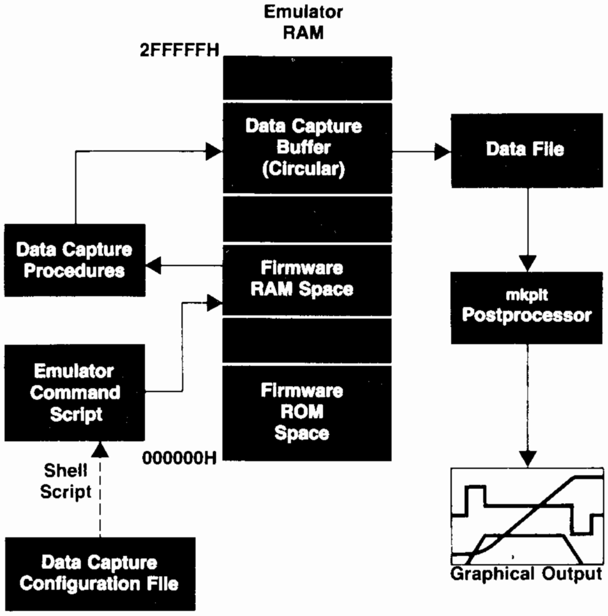


Fig. 1. *Operation of the data capture system.*

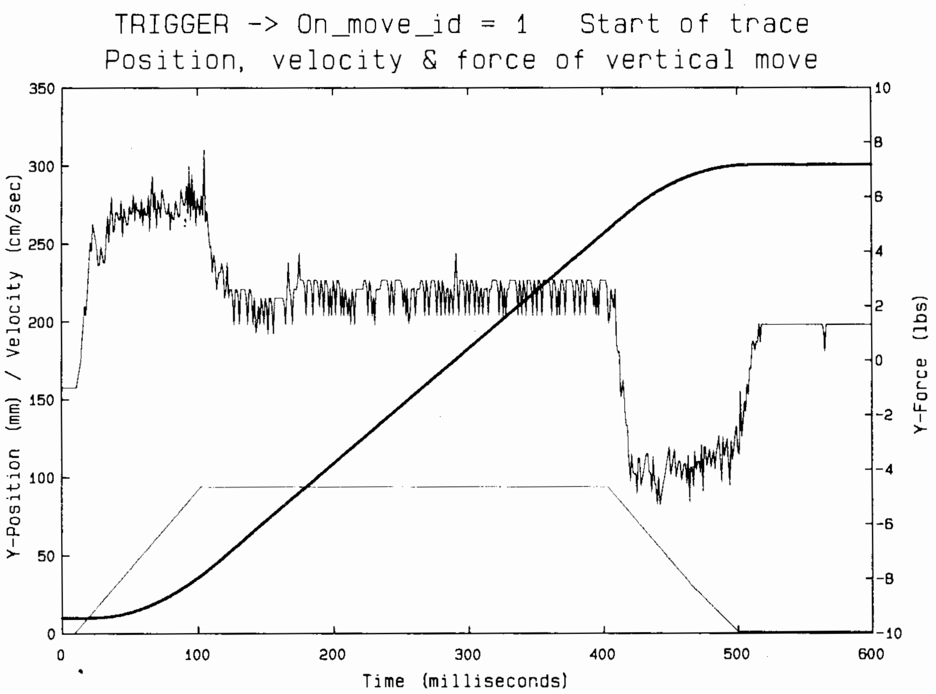


Fig. 2. *Typical data capture system plot.*

data, and then plotting the resulting data file. A trace of different variables triggered by a different event can be captured easily by simply editing a new configuration file and repeating the aforementioned process.

Fig. 2 is a plot generated by the data capture system. The data for this plot was captured during a vertical motion of the autochanger's transport mechanism. It shows the position, velocity, and force applied by the vertical motor.

Second System

A second data capture system was developed that further exploits the hardware on the autochanger controller board. This system collects a few vital bytes of data every 10 milliseconds and transfers this data to a workstation for collection and postprocessing. The controller board contains an RS-232 port, which can be connected to a terminal for debugging purposes. The design team determined that this interface was capable of transferring 18 bytes of data every 10 milliseconds if the baud rate were set at 19,200 baud. Using this information, the designers decided upon a select number of important variables that would be most useful in deciphering and debugging error conditions. Firmware was written that gathers these bytes of data and transfers them to the RS-232 port upon demand.

Data collection is accomplished by first establishing a physical connection between the autochanger under test and an HP-UX workstation via an RS-232 port. The **cu** program (**cu** is a standard HP-UX communication program) is invoked on the workstation and its output is redirected into a file. When the appropriate command is typed, the autochanger firmware begins transferring the 18-byte packets of data to the workstation. This data collection may last from seconds to hours, depending on the objective of the autochanger testing. The output file can be processed concurrently with the data collection or examined after the data transfer is completed.

Two programs were developed to process the data produced by this system. The first, called **pll**, is used to filter out any incomplete packets of data. The first byte of data in each packet is a counter, which is incremented after each packet is transferred. This is used by **pll** to screen out any data dropouts. The second program, **mcsplt**, is a postprocessing program similar to the one used in the emulation-based capture system. However, this program offers many more triggering and display features. With **mcsplt**, it is possible to scan through vast amounts of data and plot only the specific condition or conditions that are requested. Statistical values, such as minimum, maximum, and mean force measured over minutes or hours, can be plotted quickly. A weekend's worth of data, collected from an operating autochanger, can be easily scanned to locate and examine the events that led to a specific error code. The usefulness of this tool in debugging infrequent error conditions cannot be overestimated.

Acknowledgments

Autoplot was written by Bob Jewett of HP Labs. Jeff Kato wrote the data processing programs **pll** and **mcsplt**.

Mark Bianchi
Development Engineer
Greeley Storage Division

ployed in conjunction with firmware algorithms to calibrate the transport system.

The autochanger is designed so that all magazine slots and drives are accurately mounted and referenced to the walls of the autochanger structure. The optical sensors are also accurately referenced to this same structure and each one's trip point is very tightly specified. Hence, the distance from the trip point to any one of the vertical locations is known within a very small mechanical tolerance. The firmware contains a table of these distances stored in ROM. By measuring the height of the sensors with respect to the vertical origin, the firmware is able to position the front of the transport accurately at any storage slot or drive. Measurement of these sensors is performed by moving the transport down toward a sensor (with the plunge leadscrew facing upward) while monitoring the output state of that sensor. The state of the sensor will change when the appropriate flag on the transport sleeve interrupts the optical beam. When this occurs, the vertical encoder position is stored as the height of the sensor. A similar measurement is performed after the transport sleeve has been flipped and the plunge leadscrew is facing downward. The difference in height between the two sides of the sleeve is stored as another calibration offset. Thus, accurate positioning to a given storage slot or drive entails a combination of three distances: (1) the mechanical distance from the optical sensor to the slot or drive of interest, (2) the distance from the vertical origin to the optical sensor located in the same stack as the slot or drive of interest, and (3) an offset resulting from the orientation of the transport sleeve.

Service

It is important that the find home and calibration processes be as adaptive and fault-tolerant as possible, since proper operation of the autochanger depends upon their successful completion. If these processes cannot be successfully completed, then the autochanger is inoperable and must be serviced by a customer engineer. Unnecessary service calls negatively impact the product's reliability and add to its cost of ownership. For these reasons, much effort was spent on the design of the find home algorithms in an effort to make them as robust as possible.

In the event that something has broken within the autochanger, repair time can be significantly reduced if the autochanger can make intelligent inferences regarding the faulty components. The find home and calibration routines perform checks after each motion to determine if an incorrect condition exists. If such a condition exists, the firmware will set an appropriate error code and will suggest a list of up to three field replaceable units that it believes may be faulty. This self-diagnosis allows the customer engineer to verify and repair the faulty units rapidly.

Error Recovery

Error recovery is the process by which an unexpected condition is rectified so that normal operation can continue. In the autochanger mechanism, errors may occur for a number of different reasons. Some of these reasons are:

- The host computer requested that a cartridge be moved from a location that did not contain a cartridge.
- There is a temporary mechanical misalignment because of the dynamic nature of the mechanism.
- A power failure has occurred during a movement.
- There has been a mechanical or electronic failure of the autochanger.

The error recovery firmware is designed to recover from a plethora of different error conditions and provide accurate information to the host computer regarding the status of the mechanism.

The four functions of the error recovery routines are error prevention, error detection, restoration of the mechanism to normal operating conditions after an error is detected, and completion of the command during which the error occurred. Error prevention involves verifying that the requested source location contains a cartridge and that the requested destination location is empty. This is done by examining an array in nonvolatile RAM that contains the status of each element within the autochanger. If the firmware believes that the requested move would result in an error, the command is rejected before any motion is attempted. This method of prevention, although very reliable, is not foolproof. The element status array may be incorrect if a customer engineer manually moves cartridges around within the autochanger without reinitializing the element status array. In this event, the firmware must rely on the second facet of error recovery, namely error detection.

Error detection is the means by which the firmware determines that something out of the ordinary has occurred within the autochanger. The firmware detects errors in two ways. The first involves the servo loop monitors, which run continuously during autochanger motion. These routines monitor the forces that are being exerted by the vertical and plunge axes. If either of the measured forces should exceed levels specified for a given motion, the monitor routines immediately disable the servo system and set an error flag.

The second method of detection involves the use of force sense of touch at key positions during cartridge movement. While a cartridge is being extracted from its storage slot, the firmware moves the transport thumb outward slightly beyond the point at which it should engage the cartridge. If no change in force is encountered during this move, then the storage slot is assumed to be empty and an error has occurred. Upon returning a cartridge to a storage slot, the same outward movement is performed to ensure that a cartridge was in the transport. In addition, after retracting the thumb back into the transport, a small vertical motion is performed while monitoring the force on the vertical axis. A large change in the vertical force signifies a failure by the mechanism to release the cartridge. Similar tests are performed when inserting and extracting cartridges into and from the optical drives.

The design of the error recovery firmware is predicated on the assumption that because of the simple and reliable design of the mechanics, error conditions should occur very infrequently. This means that the execution speed of the error recovery routines can be reduced significantly without negatively impacting the overall performance of the autochanger. As a result, the firmware controlling the normal operation of the autochanger is greatly simplified by partitioning the code so that all error recovery algorithms are consolidated into one functional area and all motion control firmware resides in another. A simplified hierarchical diagram of the motion control firmware and error recovery firmware is shown in Fig. 4.

The motion control firmware is designed to complete its execution regardless of the status of the servo system. The lowest-level procedures that perform motion control test the status of the servo system. If the servos have been disabled, the procedures simply return and no motion is performed. All attempts to modify mechanism state information occur via procedures that verify the status of the servo system. These state variables will not be modified if the servos have been disabled. Therefore, if an error occurs, the motion control firmware completes its execution in a

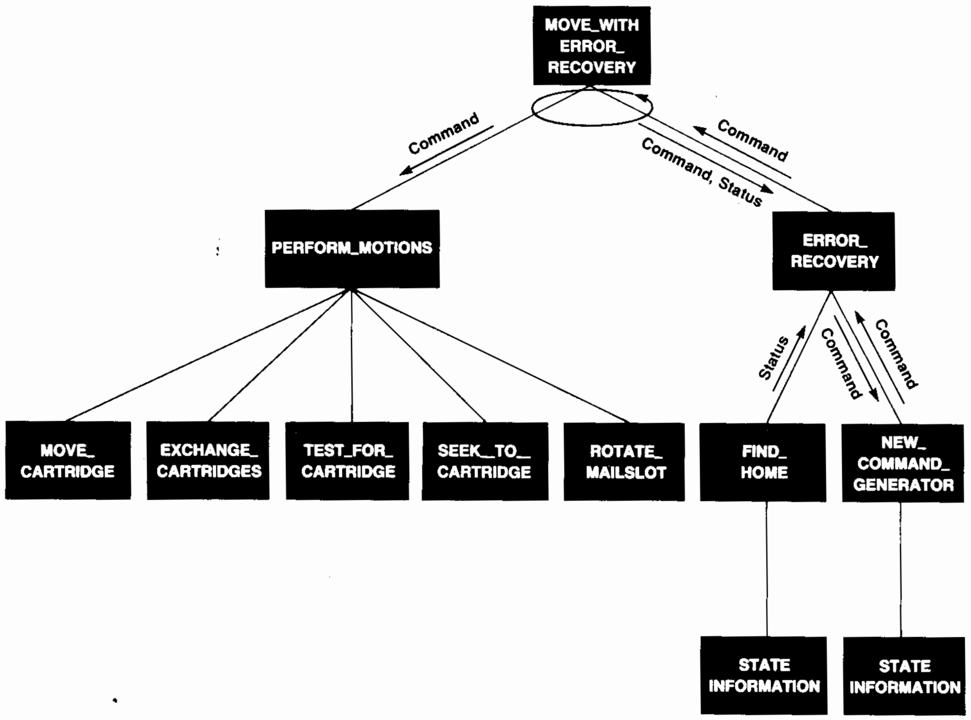


Fig. 4. *Simplified hierarchical diagram of the motion control and error recovery firmware.*

normal manner, but no motion occurs and the state of the mechanism is preserved. This provides the error recovery firmware with a snapshot of the mechanism at the instant that an error occurred. Error recovery can then proceed with restoration of the mechanics.

As can be seen in Fig. 4, ERROR_RECOVERY is composed of the two procedures: FIND_HOME and NEW_COMMAND_GENERATOR. These two procedures can be thought of as physical recovery and logical recovery procedures, respectively. The FIND_HOME procedure is responsible for successfully maneuvering the mechanism out of the error condition and then restoring the mechanics to a known initial state. It makes extensive use of the snapshot of state information that was preserved when the error occurred, and of information gathered via force sense of touch movements. FIND_HOME is the same procedure that is used for power-up initialization. Therefore, once it completes, the mechanism is in a known state and safe motions can be performed. FIND_HOME is invoked by ERROR_RECOVERY whenever a physical error has occurred that forces the servos to be disabled. The logical recovery segment of ERROR_RECOVERY is dependent upon FIND_HOME's successful completion. If FIND_HOME does fail, then ERROR_RECOVERY assumes that something is drastically wrong and calls a routine that attempts to diagnose the failure of the mechanics.

The NEW_COMMAND_GENERATOR procedure is composed of a number of routines that perform three main tasks. The first is to examine the original command and the current state of the machine. The second is to generate a new command that will either gather more information, attempt to complete the original command, or attempt to restore the autochanger to the configuration that existed just before the original command was issued. The third task is to send the newly formed command back to the PERFORM_MOTIONS routine to be executed. This process may be repeated a number of times to complete a command or restore the autochanger successfully. This looping process is initiated by any movement error that occurs while PERFORM_MOTIONS is executing. Once in this loop, ERROR_RECOVERY is directed by a state machine which determines how to resolve the error condition or exit gracefully. Each type of motion command (move cartridge, exchange cartridges, test for cartridge, seek to cartridge, or rotate mailslot) has a state machine sequence that is designed to solve the specific recovery requirements of that particular motion type. However, all state machines are composed of four common states: error recovery initialization, retry the original command, restore the autochanger to its original configuration, and return a pass or fail status. A detailed state diagram for the move cartridge error recovery algorithm is given in Fig. 5.

An error that occurs during a move cartridge command will cause ERROR_RECOVERY to begin execution. The error recovery state machine is set to its initial state, during which the original move command is stored for later use and FIND_HOME is invoked to return the mechanism to a known state. FIND_HOME will then return a pass/fail status and will provide information regarding the presence or absence of a cartridge in the transport. The state machine changes to the Test_Source state, during which a test for cartridge command is generated. This command is passed back to the PERFORM_MOTIONS procedure, which will execute a physical test of the source location for the presence or absence of a cartridge. The result of this test is passed back to ERROR_RECOVERY. If a motion error occurs during this test, ERROR_RECOVERY will invoke FIND_HOME to resolve the error and the same test for cartridge command is repeated. This process can be repeated a number of times, and if motion errors continue to occur, the state machine will decide that error recovery has failed. However, if no

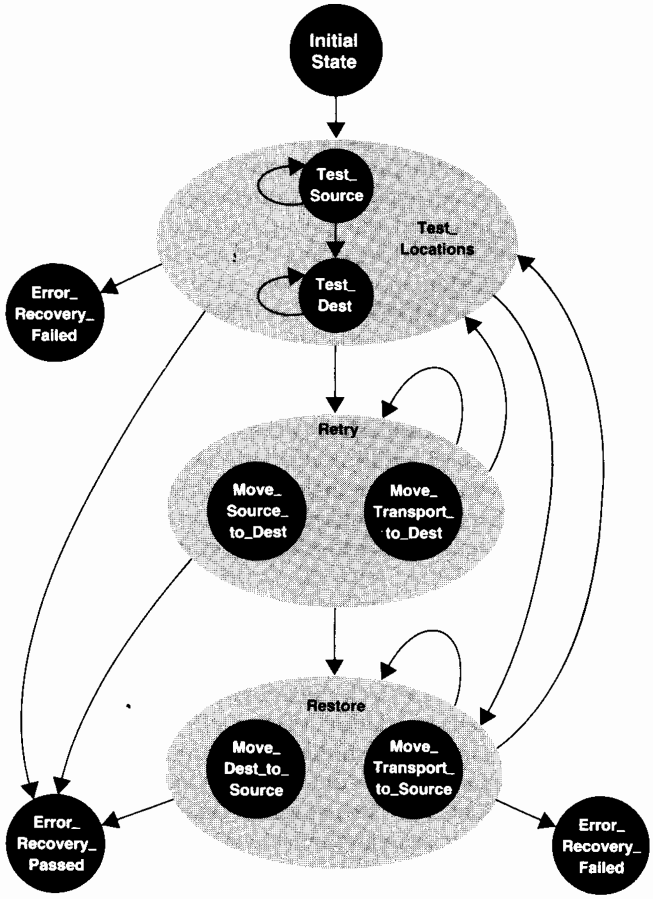


Fig. 5. *State diagram for the move cartridge error recovery state machine.*

motion error occurs, the state machine proceeds to the Test_Dest state, during which another test for cartridge command is generated and passed back to PEFORM_MOTIONS. The destination location is then physically tested and the result is passed back to ERROR_RECOVERY. As with the Test_Source state, any motion error will cause FIND_HOME to be executed again along with a reexecution of the test for cartridge command.

At this point, the state machine knows whether the source, destination, and transport are full or empty. Any full/empty combination that should not logically be possible (such as the source being empty, the transport being full, and the destination also being full) causes error recovery to fail and an appropriate error status is returned to the host. If the source is empty, the transport is empty, and the destination is full, then the command must have completed and the state machine proceeds to the Error_Recovery_Passed state. Otherwise, the state machine proceeds to the Retry state and the appropriate move command is generated. This command is either a move from the original source to the original destination or a move from the transport to the destination.

The command is once again passed back to PERFORM_MOTIONS and a cartridge movement is attempted. If it is successful, the state machine proceeds to the Error_Recovery_Passed state. Otherwise, FIND_HOME is again invoked and

Error Injection

Error recovery is a very complex procedure, as explained in the accompanying article. The number of possible situations from which the autochanger must recover is very large. To induce these errors physically would have required many engineer-hours that the development team didn't have. In addition, many errors are extremely difficult to induce. Since the error recovery testing had to be repeated every time the error recovery firmware was changed, it was deemed necessary to have error injection built into the product for the purpose of testing error recovery.

The error injection facility is enabled via the SCSI so that tests can run automatically. It can inject errors for any move at any position. Multiple errors can be queued. The facility injects errors at the lowest possible level for maximum firmware testing. It can also simulate power failure.

The built-in error injection firmware can be divided into two major functions: setting up the error trigger and injecting the error.

Setting up the Error Trigger. The objective of error injection is to be able to inject all possible errors for any move at any position on the vertical or plunge axis. All motion is broken down into many submoves. Each submove is assigned a unique move ID. When the SCSI command is sent to enable error injection, all the pertinent information needed is sent with that command to set up the trigger conditions. The injection information includes the move ID to trigger on, the axis to trigger on, the position to trigger on, and the type of error to inject.

Injecting the Error. Once the error injection command is sent over the SCSI, the built-in error injection firmware is then armed and waits for the trigger conditions to be met. When the trigger conditions are met, the error will be injected. Three types of errors can be injected:

- Servo monitor error. The servo monitor status is intercepted and an injected status is substituted. The injected status may be overforce, overcurrent, or overvoltage. This simulates unexpected physical errors.
- Force sense of touch error. The status returned from the force sense of touch is intercepted and an injected status is substituted. For example, when the autochanger is expecting to feel a cartridge, an error can be injected to tell the autochanger that it did not feel the cartridge.
- Powerfail error. When the trigger conditions are met, the built-in error injection firmware simply jumps to the power-up vectors. This makes it possible to test powerfail operation automatically for all situations.

Automatic Testing

Having error injection capability was only the beginning. The next step was to develop test suites that could could be run from an SCSI host to test the autochanger automatically. Developing these tests was the major part of the overall error injection testing development. The test suite not only had to send the appropriate injection commands to enable error injection, but also had to be smart enough to issue the correct SCSI move command to trigger the injected error, and to check for the proper SCSI status. The following test suites were developed:

- Inject errors for all possible move IDs
- Inject errors at predetermined risky positions
- Inject errors at random positions
- All of the above with powerfail errors injected.

Rick Kato
Development Engineer
Greeley Storage Division

the state machine returns to the Test_Locations sequence to determine which locations are full or empty. Once the Test_Source and Test_Dest functions have completed and returned the status of both locations, the state machine cycles back to the Retry state and either a Move_Source_to_Dest or a Move_Transport_to_Dest operation is initiated, depending on the presence or absence of a cartridge in the transport. This retry sequence may also be repeated a number of times if motion errors continue to occur, and then the state machine will proceed to the Restore state.

A very similar chain of events occurs in the Restore state, except that in this state the exit criterion is the successful restoration of the autochanger to the configuration that existed before the original command was attempted. The move commands that are generated will attempt to place the cartridge back into the source location. As with the Retry state, any movement errors that occur during an attempted restore will cause the state machine to cycle between the Test_Locations sequence and the Restore state. This process may be repeated a number of times, after which the state machine arrives at the Error_Recovery_Failed state. Once the state machine is in either the Error_Recovery_Passed or the Error_Recovery_Failed state, the appropriate status information is returned by MOVE_WITH_ERROR_RECOVERY to the host and error recovery is complete.

Firmware Development Environment

A number of factors contributed to the successful development of the autochanger firmware. One significant factor was the use of individual workstations connected by a local area network (LAN). This networking provided independent access to a common set of source code. The code files for the project resided on one hard disc, and each development workstation used the Network File System (NFS)* to gain access to the files. Each workstation had the same view of the source code, but each firmware designer was able to work independently on an HP 9000 Model 370 workstation. The bottleneck normally produced by editing, compiling, and linking on one machine was removed. In addition to increasing the designers' productivity, the use of a common set of code files facilitated revision control, tool development, and system administration.

Another factor that greatly contributed to the autochanger program was the development of the data capture tool (see box, page 29). This tool provided a means for capturing and displaying any time-varying global variable within the firmware. Data capture was primarily used to focus on mechanical or servo aspects of the autochanger, such as the position, velocity, or force of one or both axes of motion. It provided the designers with a "mechanical oscilloscope" that produced enlightening views into the operations of the autochanger. Hence, it was extremely useful in examining and diagnosing operational errors.

While data capture provided useful insight regarding errors that occurred during the autochanger development, another tool provided the means for exercising the error recovery code without requiring that specific errors occur. The error injection tool (see box, page 33) allowed the firmware designers to force an error to occur during a specific motion and at a specific position or range of positions. By using this tool, the many different states of the error recovery code could be debugged, tested, and verified. Since an error can occur during any portion of the mechanism's operation, simulating all errors by physically inducing an error would be extremely difficult to do. Error injection solved this problem and provided a powerful and flexible means for ensuring the reliability of the error recovery firmware.

A third tool that contributed to the firmware development was the mechanical regression test suite. This set of procedures provided the means to measure various mechanical aspects of each unit. Friction tests, spring constants, hard stops and datum, performance measurements, and servo parameters could be acquired using these utilities. Measurements could be taken, then tests run and the measurements rerun to assess various factors (degradation over time or temperature, effectiveness of a part or design change, baseline measurement). These routines are now used during the manufacturing process to assess the correctness of the unit's assembly.

Acknowledgments

The authors wish to acknowledge the efforts of the other two firmware designers: Rick Kato and Kraig Proehl. All four designers contributed to the design, implementation, and testing of the autochanger firmware. Special thanks to the management team that helped make the autochanger a reality. The team includes section manager Don Stavely, mechanical design manager Mark Wanger, and electrical hardware and firmware manager Mark Gembarowski. Special thanks to Colette Howe for her assistance in testing, tracking, and debugging the firmware. Additional thanks to Jeff Kato, Joel Larner, Kelly Reasoner, John Schere, and Myron Yoknis for their assistance in refining the autochanger firmware. The authors would also like to thank the R&D and manufacturing teams from HP's San Diego Division who offered their expertise in plotter technology.

---

*Network File System is a product of Sun Microsystems.

Qualification of an Optical Disk Drive for Autochanger Use

Ninety-three design changes were made to the stand-alone drive to qualify it for use in an autochanger

by Kevin S. Saldanha and Colette T. Howe

IN ADDITION TO THE USUAL requirements of data integrity, performance, and reliability for a mass storage device, an optical disc drive that is to be used in an autochanger requires a well-defined communications and mechanical interface that operates efficiently and reliably over hundreds of thousands of load and unload cycles.

In designing the HP Series 6300 Model 20GB/A rewritable optical disk library system, we had control over the autochanger end of this interface, but we had to work closely with the drive vendor to establish the other end. The vendor's original design goals had not included use of the drive in autochanger environments. Endowing the drive with this additional functionality and verifying it proved quite a challenge. Between the release of the stand-alone product (Model 650/A) and the autochanger product (Model 20GB/A) there were 93 changes to the drive.

Communications Interface

We needed a reliable interface to the drive that did not impair performance by tying up the SCSI bus. During loads, the autochanger needs feedback from the drive to know when the drive has accepted a cartridge that has been inserted. The autochanger also needs to be able to initiate ejection of a cartridge from the drive, and during operation and under fault conditions, the drive status needs to be communicated to the autochanger controller.

This interface is implemented with two hardwired signal lines: an eject line and a busy line. The semaphore on the busy line indicates drive status, fault conditions on load, and the acceptance of the cartridge in the drive. The timing of the signals was worked out based on a thorough understanding of the load and unload sequences and retry algorithms in the drive. As a result of this effort, we were able to provide inputs to the drive vendor that made these processes shorter and more reliable.

The spin-up sequence during loads includes a read of prestamped control tracks. Part of this is phase-encoded information (PEP) that is read before tracking is established, and can take up to 1.5 seconds to read. The autochanger requires this information just once, when the cartridge is first introduced into it. To eliminate subsequent PEP reads, a third hardwired signal line is included.

Mechanical Interface

Except for not using the eject button on the drive, the mechanical interface definition is no different from that in a manually operated drive. This is largely because of the flexibility of the autochanger architecture. Once we obtained specifications on acceptable insertion windows, angles, and distances, it was possible to program the autochanger to operate within these limits. The main requirement the autochanger has of the drive is that the cartridge eject to a certain minimum distance and with a specified minimum force.

Design Goals

The metric chosen to gauge the reliability of the drive in load/unload cycling was the mean number of swaps between failures (MSBF). The drive as released for use in stand-alone operation had an MSBF of 5,000, which is adequate for manual use. However, this fell far short of the target of 40,000 set for the release of the drive for the autochanger product. With the numerous changes implemented in the drive and media, we were able to prove an MSBF of 150,000 with a confidence level of 95% at the release of the autochanger product. For these products, the drive was operating in the normal horizontal mode. Work is currently underway on a project that will also allow operation of the autochanger and the drive in it on their sides in vertical mode. We have already achieved an MSBF of 200,000 in both these axes (Fig. 1).

Test Strategy

The ideal test vehicle was, of course, the autochanger itself. We could test the drive, the autochanger, and the interface in conditions similar to actual use. Indeed, this is how the bulk of our testing was conducted, and much valuable information was learned from it. As we began finding and fixing the more obvious problems, it took longer to find the more intermittent and wear dependent ones. It takes about two months to perform 200,000 swaps in an autochanger doing nothing else.

We developed special test fixtures to perform specific tests at much higher rates so we could accelerate the test cycle. One of these, the "scrubber," could insert and withdraw a cartridge in a loader tray and complete 200,000 cycles in a week. We also developed a drive tester that essentially was a stationary autochanger picker, which we used to test drives and cartridges. We also provided one of these to the drive vendor so they could duplicate and better understand problems that were encountered.

We maintained detailed records of failures encountered by the autochanger, drive, and cartridge. It proved useful to keep the swap count of a cartridge on the disk itself. A list of all known problems and their resolution status was also maintained and updated regularly. It served as a useful

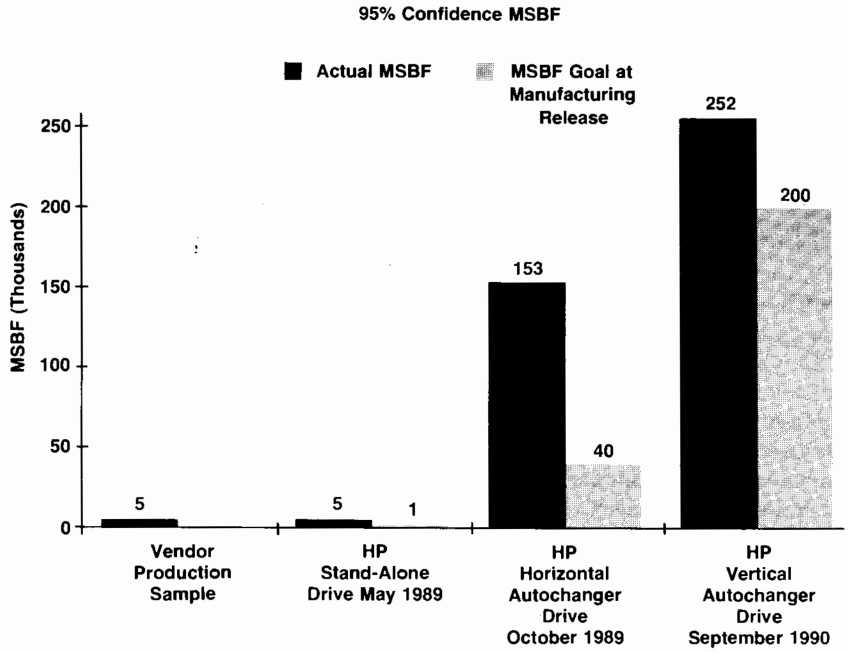


Fig. 1. *Reliability improvement in the optical disk drive. MSBF stands for mean swaps between failures.*

measure of how the qualification effort was progressing (Fig. 2).

Problems and Solutions

We were able to identify and solve problems in several areas as a result of this testing.

The early force-distance profiles used by the autochanger during inserts resulted in excessive loads at first contact with the cartridge slot door and misdetection of the hard stop in the loader tray. This force profile was tuned to deliver a much lower force to the cartridge and to accommodate significant variations in the location of the hard stop (see article, page 14).

We encountered bus hangups stemming from noise on the SCSI and the autochanger-to-drive interface lines. Cables were shortened and rerouted away from noise sources to eliminate these problems.

Some of the misload failures observed were attributable to misdetection of signals at the communications interface between the drive and the autochanger. Changes in timing and debouncing of signals both in the autochanger and in the drive made this detection far more reliable and fixed

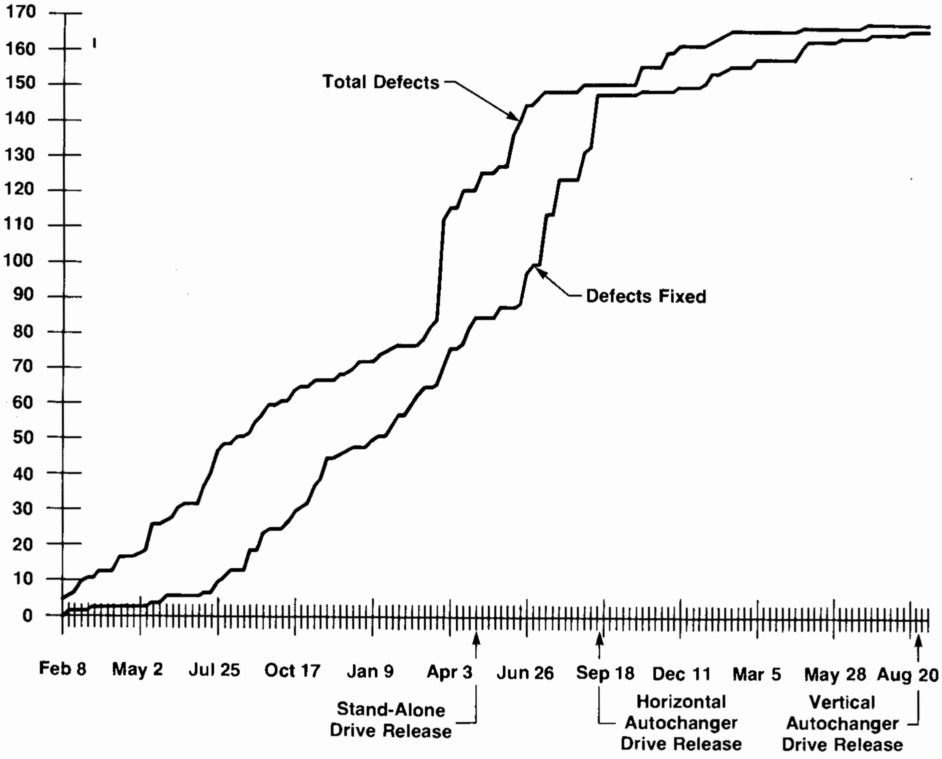


Fig. 2. *A count of known problems was a useful measure of the progress of the drive qualification effort.*

these problems.

Mechanical Problems

Several mechanical problems were encountered and fixed in the course of this development effort. Many of these were related to friction wear between contacting surfaces.

The cartridge, then made of ABS, was subject to heavy wear where it made contact with the autochanger picker, its shutter, the slot door, the loader tray, and the alignment pins in the drives. The ABS dust generated posed a potential threat to data integrity if it landed on the disc or the objective lens. More immediately, these dust deposits caused the cartridge to bind in the loader tray or catch on the alignment pins. Measures taken to minimize this include coating the loader tray with Teflon, blunting sharp lead-in angles on the tray, increasing the radii of the locator pin tips and changing the slot door material to polycarbonate.

Dust from wear was also generated within the cartridge from contact with the disk. This dust was deposited directly on the disk during unloads, when the media spins down onto landing pads on the cartridge. This contamination, which started out as a ring at the inner diameter, could migrate out over the control tracks and the data areas, and could even find its way onto the objective lens. With as few as 8,000 loads into a drive, an ABS cartridge could cause problems such as failure to spin up successfully because of obscured control tracks, improper focusing, and increases in raw byte error rates. These wear problems were overcome by changing the cartridge material to a polycarbonate and by gluing a slip ring onto the disk where it comes into contact with the landing pad on the cartridge.

There was also wear between the drive spindle motor shaft and the the disk hub. Wear on the hub resulted in a sharp edge, which at some point seized the spindle during loads and unloads. These problems were solved by modifying the spindle and hub geometry to make the capture easier and by increasing the spindle hardness. A disk leveling pad was designed to keep the disk from being loaded and unloaded at an angle. The magnetic chuck that holds the disk was modified to provide a stronger capture force-distance profile. Further improvements along these lines were made for operation of the drive on its side. The strength of a retention spring was increased to compensate for the loss of the assistance of gravity in the secure seating of the cartridge against the media sensors.

In addition to mechanical wear problems, we encountered a challenging electrostatic discharge problem. The cartridges were building up a charge as they were being inserted in drives and magazines and moved around by the picker. The cartridges would discharge to the drive loader tray. While most of the charge dissipated through chassis GND, it sometimes managed to cause glitches in the drive electronics. Reproducing these problems and testing fixes was greatly helped by the media testers in use here and at the vendor's site. Although alternative conducting materials were investigated for the cartridge, the solution to these ESD problems was found in establishing a better ground path and putting low-pass filters on susceptible drive controller signals.

In the course of improving the load/unload reliability of the drive mechanism, the testing conducted was also beneficial in increasing the margin for the basic read/write functionality of the drive. The drive accesses control tracks and reads them only during spin-up. These control tracks contain preformatted data. It is harder for the drive to maintain tracking and focus on control tracks than on data tracks. Intermittent tracking problems were observed when accessing these tracks during loads in autochangers. The changes that have been implemented to fix this have resulted in overall improvements in tracking performance in the drive.

Open Communication

Effective exchange of information has been a key ingredient in the success of this effort. The qualification of the drive mechanism and controller for use in manually operated stand-alone operation began at HP's Disk Memory Division in Boise, Idaho. After evaluating several vendors, one was chosen to supply us with both drives and media. A second media source was also identified. We began working with them to define and qualify the drive and media. This effort drew on the expertise of the program in Boise.

The project was then transferred to HP's Greeley Storage Division in Colorado, where work had already begun on integrating the drive into a stand-alone product as well as into an autochanger. The flow of information between us and our vendors had by now swelled to a torrent with facsimile messages flying back and forth daily. Despite the fact that the drive vendor was in Japan, we were able to evolve a very effective working relationship. Aside from regular management contacts, we had periodic technical meetings both here and in Japan. These served to establish engineer-to-engineer contacts. It then became easy to direct communications to the right people and obtain quick resolution of issues. We were fortunate enough to be working with a responsive vendor and it was not uncommon to send a facsimile request for information one evening and return the next morning to find a response to it.

Acknowledgments

Terry Loseke provided much valuable guidance in integrating the new optical recording technology into our products. He has also championed our interests at the ANSI and ISO standards committees. Thanks also go to Brett Mortensen and Sue Magenis for their early work on the drive, to Dave Bradley and Steve Johnson for their help with test systems, to the optical group at HP's Disk Memory Division and in particular Dave Campbell for providing much insight and advice with respect to optical recording, to Ed Sponheimer who managed, and Alfred Natrasevschi, Karen Klemm, Bob Proctor, and Norm Carlson for helping to establish the stand-alone drive, to Tom Oliver for his work in debugging the drive autochanger interface and Al Piepho for his efforts on firmware testing, to the autochanger controller group including Kraig Proehl for the design of the three-wire interface and Mark Bianchi for the drive interface code, and to the materials engineering team of Mark Maguire, Steve Castle, and Mark Davis for their valuable support. Special thanks to the design team at our vendor which included Tadashi Otsuki, who managed the project and Takashi Naito, Hiroyuki Shinkai, and Yoshimori Yamasaki, who were lead engineers.

Authors

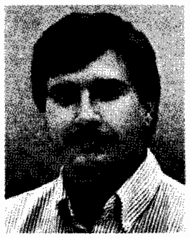
December 1990

6 Optical Library System

Donald J. Stavely

Don Stavely is an R&D section manager at HP's Greeley Storage Division. He holds a BSEE degree (1975) and an MS degree in computer, information, and control engineering (1976) from the University of Michigan. Don is currently section manager for the design of optical drives and media. Before attaining that position, he was project manager for the HP Series 6300 Model 20GB/A rewritable optical disk library system. Originally from Pontiac, Michigan, he currently lives in Fort Collins, Colorado. His favorite pastime is windsurfing.

Kraig A. Proehl



Kraig Proehl is a member of the project team that designed the firmware and hardware architecture of the Model 20GB/A autochanger. He has been an R&D engineer with HP's Greeley Storage Division since 1983. A graduate of Purdue University, he holds a BSEE degree (1983). His past projects include design work on the HP 7974 and 7980 tape drives. He is named an inventor for a patent on the HP 7980 tape drive. Kraig grew up in Trotwood, Ohio. He lives with his wife and three children in Loveland, Colorado and enjoys sports, woodworking, and photography.

Mark E. Wanger

Mark Wanger is an R&D project manager at HP's Greeley Storage Division. He was project manager for the HP Series 6300 Model 20GB/A rewritable optical disk library system. He has done mechanical and electrical engineering on several tape drive products, and his work has resulted in six patents for tape drives and magnetic-tape heads. Mark began his career with HP in 1980 at Desktop Computer Division. His educational background includes a BSME degree (1979) from the University of California at Santa Barbara, an MSME (1980) from the Massachusetts Institute of Technology, and an MSEE (1984) from Colorado State University. He was previously with Hughes Aircraft Company where his focus was satellite control systems. Mark was born in Los Angeles, California and currently lives in Fort Collins, Colorado with his wife and two children. He plays volleyball and does woodwork in his free time.

14 Autochanger Mechanical Design

Daniel R. Dauner

Dan Dauner joined HP's Fort Collins Division in 1979 and currently is an R&D engineer at the Greeley Storage Division. He contributed to the design of the HP 7980 and 9144 tape drives, developed the magazine assemblies and mailslot of the Model 20GB/A autochanger mechanism, and was responsible for the mechanical integration of the various autochanger subassemblies. A 1979 graduate of the University of Texas at Arlington, he holds a BSME degree. Dan is a member of the American Society of Mechanical Engineers (ASME). His birthplace is Wellington, Kansas. He, his wife, and his two sons live in Fort Collins, Colorado. He is a public school system volunteer, coaches and plays soccer, bicycles, and runs in his after-work hours.

Jennifer L. Methlie

Jennifer Methlie is a manufacturing engineer with HP's Greeley Storage Division. She has a BS degree in mechanical engineering (1985) from Washington University, St. Louis. Jennifer was one of the design engineers on the HP Series 6300 Model 20GB/A rewritable optical disk library system, and is named an inventor in several pending patents related to that project. Jennifer lives in Fort Collins, Colorado. Her favorite pastimes include skiing, soccer, and golf.

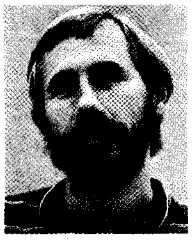
Michael L. Christensen

Mike Christensen is part of the design team for the HP Series 6300 Model 20GB/A and related products for OEM customers. He joined HP's Loveland Instrument Division in 1983, and is currently a mechanical engineer for the Greeley Storage Division. His previous job responsibilities include the design of the HP 3070 board test system at the Manufacturing Test Division, and a year as production engineer at the Loveland Instrument Division's component operation. Mike has a BS degree in mechanical engineering (1983) from California Polytechnic State University at San Lusi Obispo. He is named an inventor on a patent for the design of a fixture latching mechanism, and is a member of the American Society of Mechanical Engineers. He grew up in South Pasadena, California. He, his wife, and his son live in Fort Collins, Colorado.

Raymond C. Sherman

An R&D engineer for HP's Greeley Storage Division, Ray Sherman joined the division in 1988. He contributed to the design and development of the horizontal carriage and its supporting structures for the Model 20GB/A rewritable optical disk library system, and is involved in the development of variations of this product for OEM customers. His educational background includes a BS degree in biology (1974) from Iowa State University, an MA degree in biology (1977) from Drake University, and a BS degree in mechanical engineering (1979) from Iowa State University. Before joining HP Ray worked for Eastman Kodak as a development engineer, for ALCOA as a project engineer, and for Ball Aerospace System Division as a design engineer. He is currently registered as a mechanical engineer in the State of Iowa and is a member of the American Society of Mechanical Engineers. Born in Grinnell, Iowa, Ray is married, has two grammar-school-age sons, and lives in Greeley, Colorado. He is a hunting and hiking enthusiast.

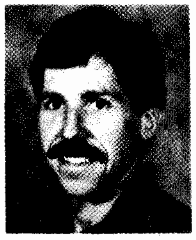
Leslie G. Christie, Jr.



Leslie Christie holds BSEE and MSES degrees from the University of Arkansas. He has been an R&D engineer for HP's Greeley Storage Division since 1984. For the HP Series 6300 Model 20GB/A autochanger, he designed the vertical carriage, mechanical drive functions, and mailslot. Previously, he designed the buffer arm assembly for the HP 7980 tape drive. His work resulted in two patents related to tape drives. Before joining HP, Leslie was an instructor at Kansas State University. Originally from Fayetteville, Arkansas, Leslie is married, has two children, and lives in Greeley, Colorado. His hobbies include gardening, golf, and tennis.

24 Autochanger Servo Design

Mark Bianchi



An R&D engineer for HP's Greeley Storage Division, Mark Bianchi has been with HP since 1984. He is a 1984 graduate of Pennsylvania State University and holds a BSEE degree. Mark designed the calibration and recovery firmware for HP Series 6300 Model 20GB/A autochangers, and was also responsible for the design and development of productivity-enhancing and analysis tools to support that project. His previous projects include designing the read/write channel electronics for the HP 9144A and 9142A tape drives. He was also responsible for the design, layout, and testing of the data compression chip for the HP 7980XC. Mark's professional interests include analog circuit design and control theory. Originally from Vineland, New Jersey, he currently lives in Fort Collins, Colorado. His outside interests include large-format black and white photography, volleyball, soccer, and scuba diving.

Thomas C. Oliver

Thomas Oliver is an R&D engineer for HP's Greeley Storage Division. He holds a BS degree in electrical engineering and computer science (1983) from Ohio State University. Tom is a member of the project team that designed the electronics and firmware algorithms that control the autochanger mechanism of the HP Series 6300 Model 20GB/A. His past projects include designing the servo systems for the HP 9144A and 9142A tape drives. Before joining HP, Tom worked for Liebert Corporation on a microprocessor-based controller design. His major professional interest is high-performance, multivariant control systems. Tom grew up in Columbus, Ohio. He and his wife make their home in Fort Collins, Colorado. His outside interests include windsurfing, weight lifting, basketball, skiing, and blues music.

35 Optical Drive Qualification

Kevin S. Saldanha

Kevin Saldanha graduated with a BS degree in mechanical engineering in 1981 from the Indian Institute of Technology, Bombay, and earned an MS in industrial engineering from the University of Oklahoma in 1984. An R&D engineer for HP's Greeley Storage Division, he is part of the project team responsible for the qualification of the optical disk drive for Model 20GB/A autochanger. Before joining HP in 1985, Kevin worked for Xidex Magnetics and authored a 1984 paper on operations research. His professional interests now center around optical recording. Originally from Mangalore, India, he is married and lives in Fort Collins, Colorado. He enjoys music, the outdoors, and travel.

Colette T. Howe

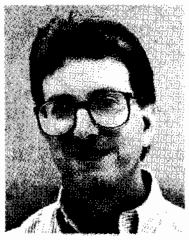
Colette Howe is an R&D engineer for HP's Greeley Storage Division. She holds a BS degree in electrical engineering (1984) from the University of Utah. She joined HP in 1984 and has contributed to the design of the HP 9153A disk drive and the design and qualification of the HP 9153B, 9123D, and 7963B disk drives. Currently, she is part of the project team responsible for qualification of the optical disk drive for the Model 20GB/A autochanger. Before joining HP, Colette was a technician with Evans & Sutherland Computer Corporation. She is a member of the IEEE Computer Society and her professional interests center around specialty drive qualifications, board layouts, optical devices, and imaging systems. She was born in Salt Lake City, Utah, and she and her husband reside in Greeley, Colorado. Her hobbies include biking, downhill skiing, windsurfing, sewing, and cooking. She also serves as Relief Society President of the Greeley University Branch for the Church of Jesus Christ of Latter Day Saints.

38 CD-ROM Drive

Edward W. Sponheimer

Ed Sponheimer joined HP's Civil Engineering Division in 1977, after receiving a BSEE degree from the University of Arizona in 1976. Ed was the project manager for the HP 9153C 40-Mbyte hard disk drive, and R&D project manager for the Model 600/A CD-ROM HP-IB drive and the Model 650/A rewritable optical stand-alone drive. He is currently the materials engineering manager for optical products. Ed came to HP from Hughes Aircraft Company, where he worked in component evaluation and analysis and test equipment engineering. Ed is a member of the Loveland Youth Baseball Associates Board of Directors. He is married and he and his wife and three sons live in Loveland, Colorado. His hobbies include baseball, golf, bowling, and the study of history.

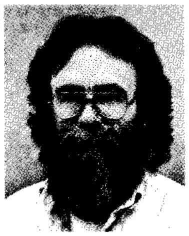
John C. Santon



Born in Grosse Ile, Michigan, John Santon graduated from the University of Florida in 1980 with a BSEE degree. He joined HP in 1981, working in the Desktop Computer Division. He has contributed to service engineering efforts on the HP 9816 computer, has done software testing on the HP 9133D and 9153A disk drives, and has done hardware design on the HP 9153B and 9153C disk drives and the Model 600/A CD-ROM drive. John is especially interested in applications of software structured design methodology to hardware design. John and his wife and two daughters live in Johnstown, Colorado. Gardening, backpacking, and radio control modeling are some of his leisure activities.

42 CD-ROM Error Correction

John C. Meyer



John Meyer received a BEE degree from the University of Minnesota in 1984, and will receive his MSEE from Colorado State University in 1990. He has been with HP since 1984. He has worked in manufacturing support for small magnetic disk peripherals. For the CD-ROM product, he developed the error correction mechanism. He is currently working on optical drive controllers, mechanism drivers, buffer management, and error correction for a new product. Before joining HP, John spent twelve years as a self-employed carpenter and contractor. He was also a U.S. Navy Seabee, working as a construction electrician. John is married, and has eight-year-old twins, one girl and one boy. He was born in Minneapolis, Minnesota, and he and his family currently live in Greeley, Colorado. He plays guitar, and enjoys rock collecting.

49 CD-ROM Software Protection

Kenneth R. Nielsen

Born in Minneapolis, Minnesota, Ken Nielsen graduated from Dunwoody Industrial Institute in 1968 as an electronic technician. After joining HP in 1969, he attended Colorado State University, majoring in electrical engineering. Ken was design engineer for the