12. THEORY OF OPERATION SECTION

12. THEORY

12.1. RAID Architecture Overview

The objectives of the RAID technology are the low cost, high reliability, and high I/O performance of disk storage devices. To achieve these objectives, this subsystem supports levels 0, 1, and 5 of RAID technologies (in this section, part of level 3 RAID technology is explained to make the outline of RAID5 more understandable). The features of the levels of RAID technologies are described below.

12.1.1 RAID0 and RAID1

RAID0 uses small-scale disk storage devices instead of conventional expensive large-scale disk storage devices for cost reduction. To increase the reliability of RAID0 devices, all disk devices are duplexed in RAID1.
Strictly speaking, RAID0 and RAID1 subsystems are not disk array systems; their performance is the same as conventional disk subsystems. Controlling these subsystems are rather simple and many RAID0 and RAID1 subsystems have been put into market by several vendors.

12.1.2 RAID3

In RAID3, a stream of data to be transferred is split and distributed into two or more disk devices (1 parity group) on a byte or bit basis (striping). This enables two or more disk devices to run simultaneously and increase the speed of data transfer between the DKC and disk drives. In addition, since parity data for the parity group is created and stored on a separate disk device (parity disk), data can easily be recovered even if a device in a parity group gets inoperative or causes a read error. This enhances the reliability of the disk storage subsystem.
Since RAID3 lacks the parallel I/O capability and it entails a long latency time because of the need to drive two or more drives at a time, it exhibits little performance in applications that process small-volume data repeatedly (transaction processing) though it shows a real advantage when running applications that process large-volume data in a single run (e.g., scientific computations).

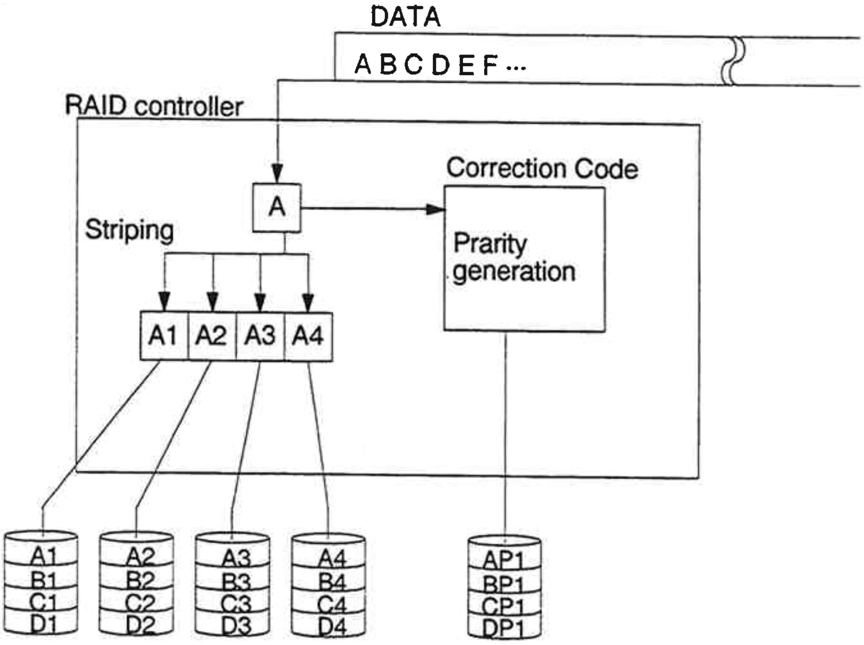


Fig. 1.2 Outline of a RAID3 (4D + 1P) System

12.1.3 RAID5

Small-sized records are intensively read and written randomly in transaction processing. This type of processing generates many I/O requests for transferring small amounts of data. In such a situation, greater importance is placed on increased I/O performance (parallel I/O processing) than on increase in the rate of transferring large-volume data. RAID5 has been introduced to be suitable for this type of transaction processing.

In RAID5, the striping size is set to that of blocks which are transferred in a small-scale I/O processing mode and which are distributed in two or more disk devices (1 parity group). This entails the RAID controller to access each disk device only for one stripe equivalence of data and allows it to perform I/O operations on other disks in parallel, increasing the I/O performance substantially; though its data transfer rate remains the same as that of conventional subsystems in small-scale I/O applications. In large-scale (sequential) I/O applications, it permits the blocks in the same parity group to be processed in parallel as does RAID3, resulting in an increase in data transfer rate. In addition, like RAID3, RAID5 uses a parity disk for improved reliability. In individual writes to single-blocks (small-scale writes), however, this parity scheme raises various problems for the following reasons:

- The parity disk entails rewrites of new parity data.
- It is necessary to read the old parity data and old (before-update data) to generate new parity data.

Consequently, RAID5 entails extra reads from the data and parity disks in small-size write operations (this is called a penalty). If the parity data were fixed at a single disk device, the parity disk would be occupied during a single write that is executed to update the parity data, thus making it impossible to perform parallel I/O processing. To alleviate this problem, RAID5 adopts a system of distributing parity data on several disks in the group. This will not solve the write penalty problem completely.

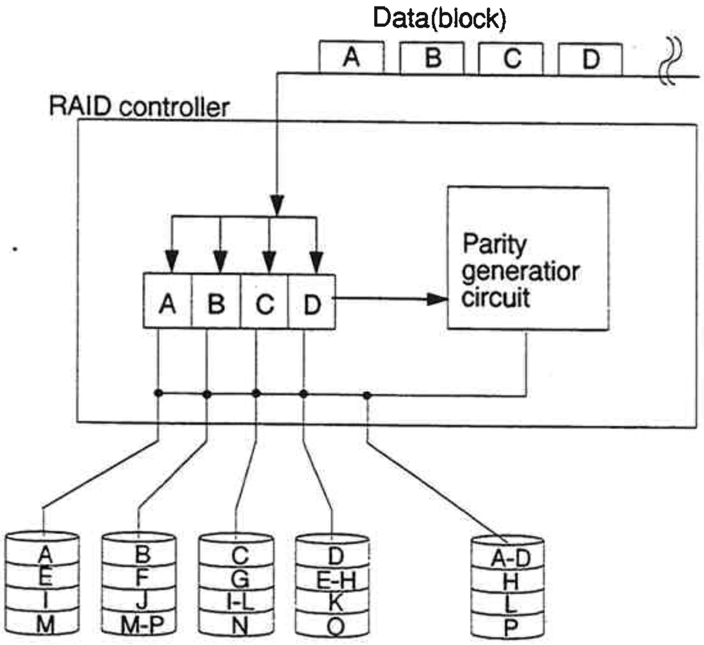


Fig. 1.3 Outline of RAID5 (4D + 1P)

12.1.4 Application of the RAID technologies DKC210 and DKU205

RAID5 will not show its stuff during transaction processing if a single I/O operation spans over two or more disk devices (when the stripe size is too small). Consequently, adequate consideration should be given to the stripe size.

In this subsystem, the stripe size is 58 KB which is equivalent to one track of the 3390-3 so that the 3390-3 can be emulated. This is because it is anticipated that a single transaction will not span over two or more tracks when the 3390-3 is emulated. this subsystem also uses cache memory to preclude write penalties from occurring wherever possible. The cache memory can pool the data to be written onto the disk drives and prefetch old data.

- On write penalty

A parity group of a this subsystem (level 5) system consists of seven disk devices (6D + 1P). Since the parity data is made up of parity data for six data disks in the group, once a 1 stripe equivalence of partial write occurs in the group during transaction processing, it becomes necessary to regenerate the corresponding parity data in that group.

Since parity data is calculated using the formula shown below, "data established before update," "parity established before update," and "data established after update" are required to generate parity. The extra processing required to read this "data before update" is referred to as a write penalty.

"New parity data"
= ("Data before update" EOR "Data after update") EOR "Parity before update"

12.2. Differences in Features between this subsystem and a Conventional System

This chapter discusses the features of this subsystem compared with a conventional system (DKC80/90).

- Employment of SCSI drives
Small HDDs are used as disk storage and WIDE SCSI is adopted as their interface.

- Disk data format altered
Since this subsystem uses SCSI drives, it employs the fixed-length FBA format instead of the variable-length CKD format as the data format.

- Necessity of cache memory
Since cache need always be passed in data transfer, conventional through operation is impossible (always DFW image). Consequently, the conventional system that uses a single cache scheme will become inoperative in case an error occurs in cache memory. This subsystem is provided with two areas of cache memory cache A and cache B) that are fully non-volatile (battery-backed).

- Cache management method
The read and write data in cache are controlled independently of each other, so that they are never overwritten as in the conventional systems.

- Employment of RISC processor
Commercially available RISC processors are adopted as the microprocessors for this subsystem.

- Independent microprograms
Each microprogram is made up of three control layers (i.e., channel command control, RAID control, and SCSI control) which run independently (asynchronously) of one another.

- IMPL system
The microprograms are loaded from the local FM on each PCB, not from the SVP HDD unlike the conventional systems.

- Correspondence between the logical and physical volumes
In RAID5(DK306-45 or DK308-45), 7 physical disks make up 8 logical volumes. In RAID5(DK308-90), 7 physical disks make up 16 logical volumes. In RAID1(DK308-90), 2 physical disks make up 3 logical volumes. In RAID5 (DK309-180), 4 physical disks make up 16 logical volumes.

- Duplex/redundancy architecture employed in major components
The major components of the subsystem (cache, high-speed I/O buses, drive paths, and AC input power supplies) are of duplex architecture. A redundancy structure is employed throughout the power supply system. These features realize high reliability of the subsystem and ensure non-disruptive execution even when errors occur in parts of the subsystem hardware.

- Personal computer employed as the SVP
The SVP is implemented by a personal computer which can run under Windows.

- Hot replace support
Components such as CHAs, DKAs, cache, disks, and power fans in the subsystem permit hot replace while the subsystem is running. Upgrade and replacement of microprograms are also allowed while the subsystem is running, thus realizing non-stop maintenance.

12.3. Specifications

12.3.1 DKC210I Specifications

Specifications are shown in the following table.

DKC210I Specifications

Item		Specifications
Data transfer rate (MB/s)		3/4.5/6.0/10/17 [Note]
Nonstop maintenance	Disk drive	○
	Control package	○
	Microcode	○
	Power supply, fan	○
	Battery	○
	Maintenance package	○
	Duplex bus	○
	Spare disk support	○

Note: 6.0MB/s data transfer can be applied on the Channel Interface when attached to Hitachi CPU.
10MB/s and 17MB/s data transfer can be applied on the Serial Channel Interface.

12.3.2 DKU205I Specifications

Specifications are shown in the following table.

DKU205I Specifications

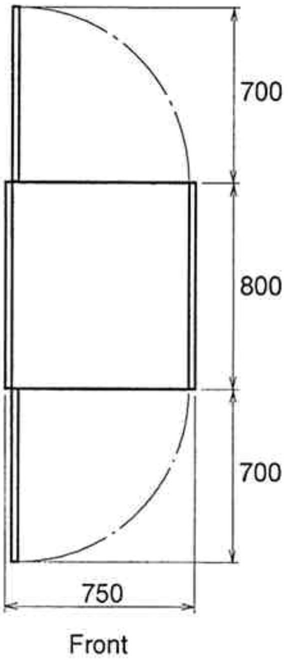
Item	DK306-45	DK306-35	DK308-90		DK308-45	DK309-180	
	RAID5	RAID0	RAID5	RAID1	RAID5	RAID5	RAID1
Seek time (ms) MIN. AVE.	2.0 11.5	2.0 10.5	2.0 12.5		2.0 12.5	2.0 12.5	
Data transfer rate (MB/s)	4.7 - 6.9	5.4 - 6.9	7.6 - 10.7		7.6 - 10.7	10.0 - 15.3	
Average latency time (ms)	4.8	4.8	4.8		4.8	4.8	
Number of physical cly./HDA (User's area only)	3,317	2,361	5,434		5,434	6,823	
Number of physical track/ physical cyl.	29	29	28		13	29	
Number of physical track/HDA (User's area only)	96,193	68,469	152,152		70,642	197,867	
Maximum memory capacity (GB/HDA)	3.784	2.838	7.568	8.514	3.784	18.463	
Number of actuator/HDA	1	1	1		1	1	
Number of HDA/Drive unit	maximum 64 (Containing 8 spare HDAs)	maximum 64	maximum 64 (Containing 8 spare HDAs)		maximum 64 (Containing 8 spare HDAs)	maximum 64 (Containing 8 spare HDAs)	

12.3.3 Physical Specifications

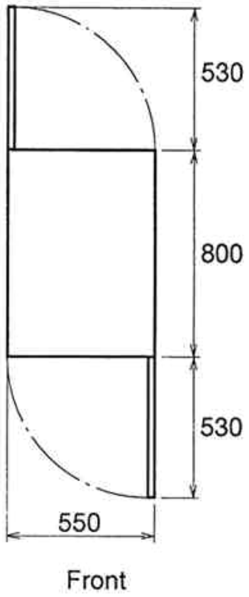
DKC210I/DKU205I physical specifications are shown in the following figures and table.

(Unit: mm)

DKC210I



DKU205I



Item		DKC210I		DKU205I
Dimension (mm)	Width	750 *1	750 *1	550
	Depth	800	800	800
	Height	1,790	1,790	1,790
Weight (kg)		310 *2	450 *3	436 *4
Heat Output (kW)		1.4 *2	2.4 *3	2.45 *4
Power Consumption (kVA)		1.5 *2	2.56 *3	2.60 *4
Air Flow (m3/min.)		12 *2	16 *3	12 *4

*1: This includes the thickness of side covers (16mm×2).

*2: These values are used when DKC210I has 4GB Cache-Memory.

*3: These values are used when DKC210I has full options.

*4: These values are used when DKU205I has full options.

12.3.4 Equipment Layout

The layout diagrams of the DKC210 and DKU205 components are shown below. For details on the individual components, see the Location Sections.

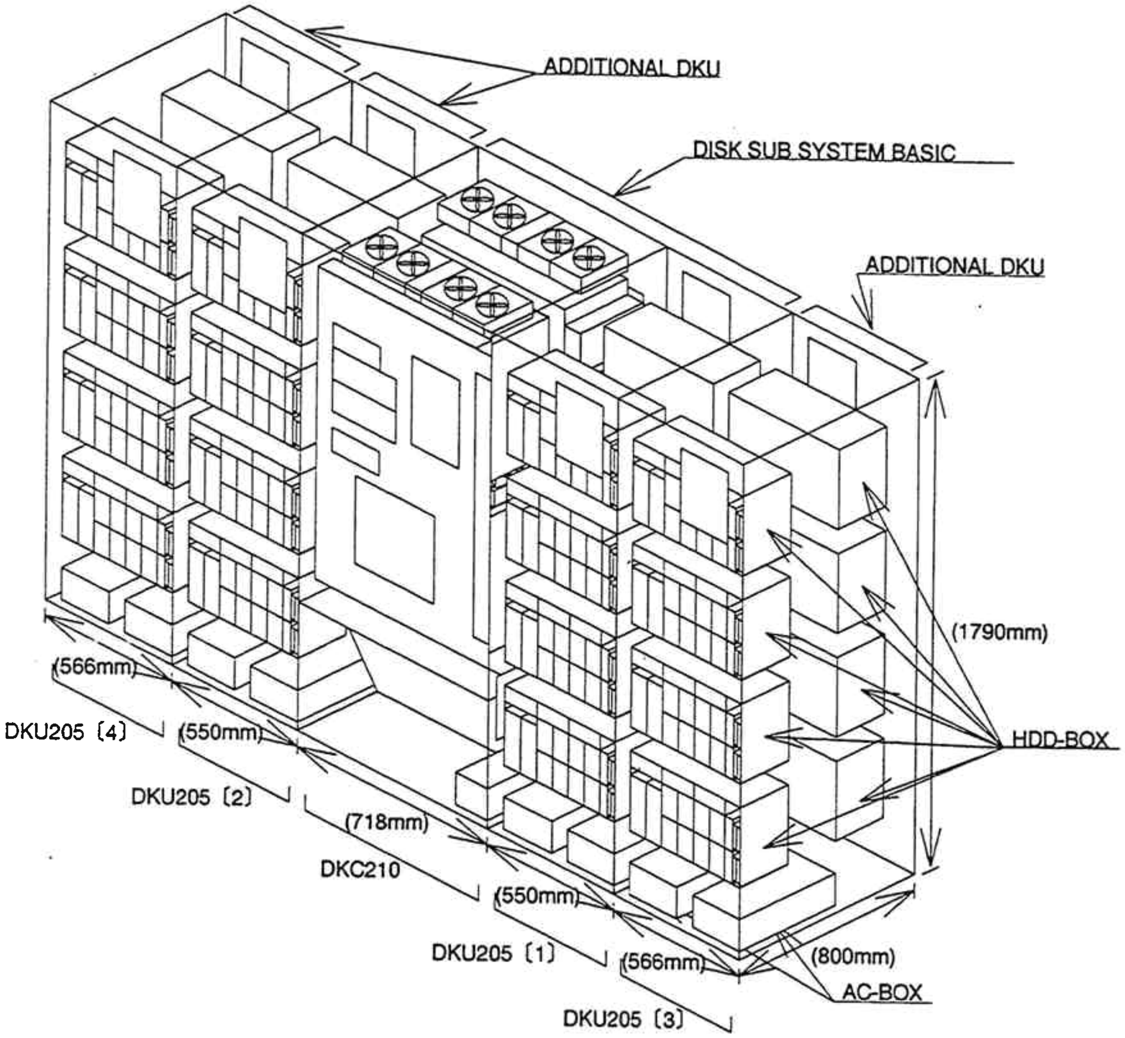


Fig.3.4 Over View of DKC210 and DKU205

12.4. Internal Operation

12.4.1 Internal Hardware Configuration

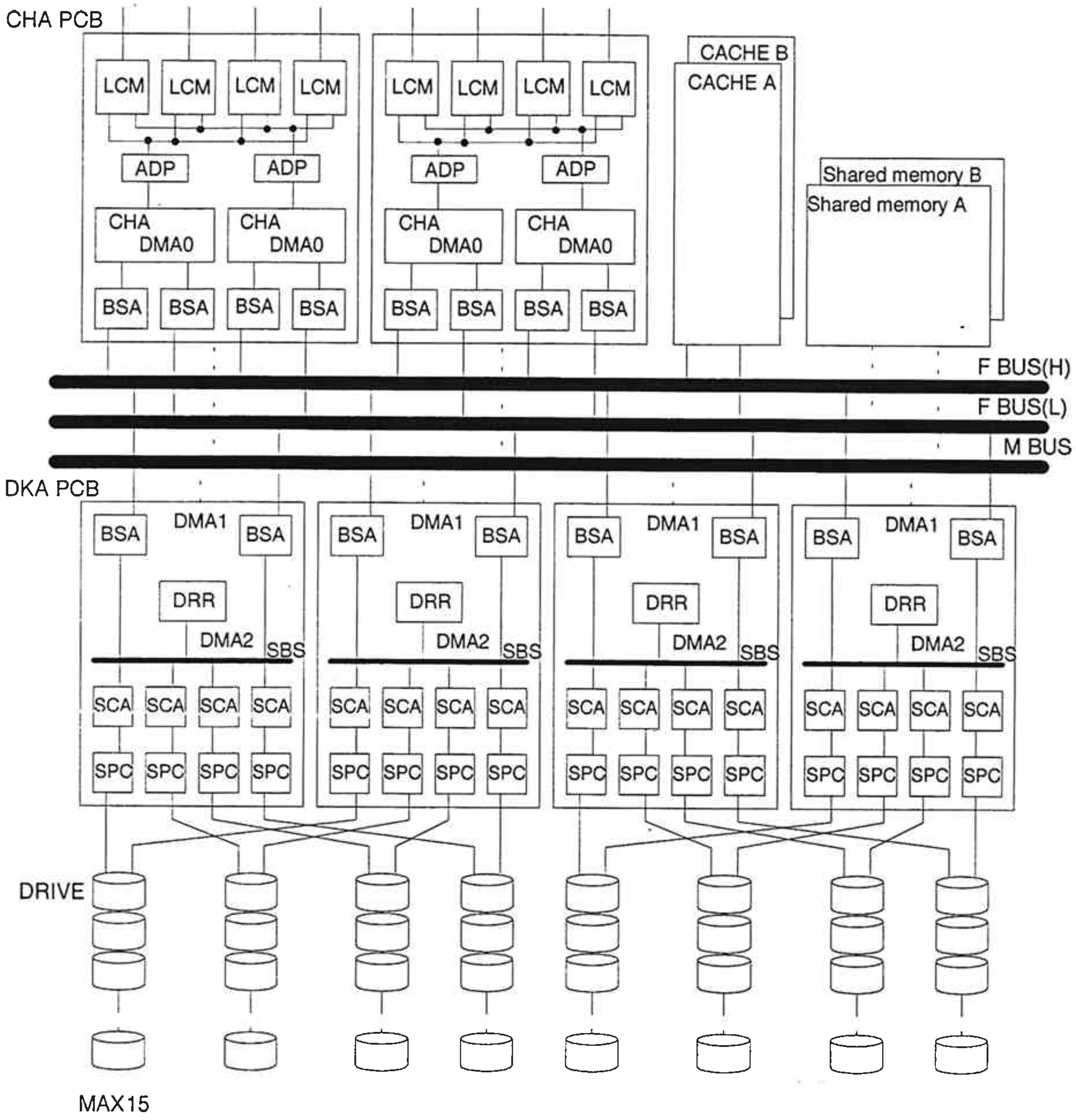


Fig. 4.1 Internal Hardware Configuration

CHA PCB (Channel Adapter Package Board):

A PCB is used to connect to the host processor. There are two types of CHA PCBs. One for optical channels and the other for metal channels. The CHA PCB for optical channels is provided with LCM in its optical link control unit.

LCM (Link Control Module):

Provides link control functions for the conversion of formats between optical and metal channel interfaces. The LCM contains an internal processor (LCP) and 2 KB of buffer memory.

ADP (ADaPter):

Hardware that connects the LCM to CHA physically and logically. The ADP contains 256 Bytes of internal buffer.

CHA (CHannel Adapter):

Provides the channel interface control functions and intercache data transfer functions. It is used to convert the data format between CKD and FBA. The CHA contains an internal processor and 128 bytes of edit buffer memory.

DMA0:

Controls the data transfer between the ADP and CACHE.

CACHE:

Lies between the channels and drives which serves as an intermediate buffer. CACHE has a maximum of 16 GB (8G × 2 areas) of capacity. It is available and controlled as two areas of cache (cache A and cache B). It is fully battery-backed (48 hours).

Shared memory:

Stores the shared information about the subsystem and cache control information (director names). These types of information are used for exclusive control of the subsystem. Like CACHE, shared memory is controlled as two areas of memory and fully non-volatile (endures approximately 96 hours). In RAID5(DK306-45), the size of shared memory must be 16 MB for 1 GB of cache. In RAID5(DK308-90 or DK309-180) or in RAID1(DK308-90 or DK309-180), the size of shared memory must be 32 MB for 1 GB of cache.

FBUS (Fast I/O BUS):

Carries the data that is transferred between adapters and cache. It is duplexed using two physical buses (H and L sides) each of which provides a data transfer rate of 200 MB per second. The L side bus is also used to access shared memory during interprocessor communication.

MBUS (Multi cpu BUS):

Used to support interprocessor communication between CHA Prog, DMPs, DSPs, and shared memory.

BSA (BUS Adapter):

Hardware that connects the PCBs, FBUS, and MBUS.

DKA (DisK Adapter):

Provides the control functions for data transfer between drives and cache. The DKA contains a DRR (Data Recover and Reconstruct), a parity generator circuit. It supports four SCSI buses and offers 64 KB of buffer for each SCSI bus. The DKA also has an internal processor.

DMA1:

Controls data transfer between the buffers in the DKA and CACHE.

DMA2:

Controls data transfer between the buffers in the DKA and drives.

DRIVE:

Represents a SCSI drive whose internal data format is FAB (fixed length).

SBS (SUB BUS):

A data bus in the DKA.

SCA (SCSI Adapter):

Controls SCSI packets.

SPC (SCSI Protocol Controller):

A SCSI initiator that controls the SCSI devices.

12.4.2 Software Organization

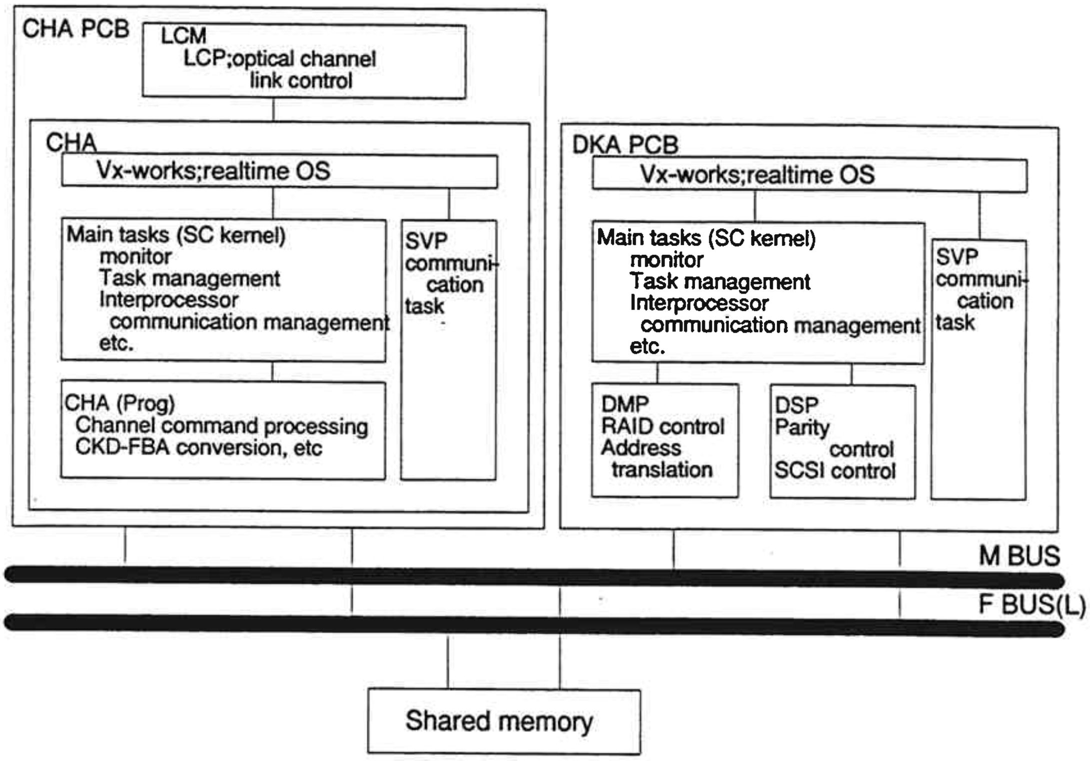


Fig. 4.2 Software Organization

Vx-Works:

A basic OS for controlling the RISC processor. Its primary tasks are to control and switch between the main tasks and SVP communication tasks.

Main tasks:

Made up of DKC control tasks (CHA Prog, DMP, DSP) and the SC kernel tasks that supervise the DKC control tasks. They switch the control tasks by making use of the SC kernel's task switching facility.

SVP communication task:

Controls the communication with the SVP.

LCP (Link Control Program):

Controls the optical channel links. LCP is identical to that which is used in the DKC90. LCP is located in the LCM.

CHA (Prog):

Is a channel command control layer that processes channel commands and controls cache and data transfer operations. It is located in the CHA. CHA Prog is recognized by the logical volume number and logical block number.

DMP (Disk Master Program):

Is a RAID control layer and provides cache control, logical-to-physical address translation, and RAID control functions. DMP is located in the DKA. DMP is recognized by the logical volume number and logical block number.

DSP (Disk Slave Program):

Is a SCSI drive control layer and provides SCSI control, drive data transfer control, and parity control functions. It is located in the DKA. DSP is recognized by the physical volume number and LBA number.

- On Interprocessor (interprogram) communication

Interprocessor communications are initiated from the MBUS via the shared memory. They are performed by the MBUS and FBUS (L) when the bus mode is the transaction mode (see "Bus Mode Switching Control"). If the MBUS is blocked, interprocessor communications are carried out via the FBUS (L). Programs running on the same processor (DMP and DSP) perform interprogram communications through program interrupts.

12.4.3 DKC210 and DKU205 Subsystem Configuration

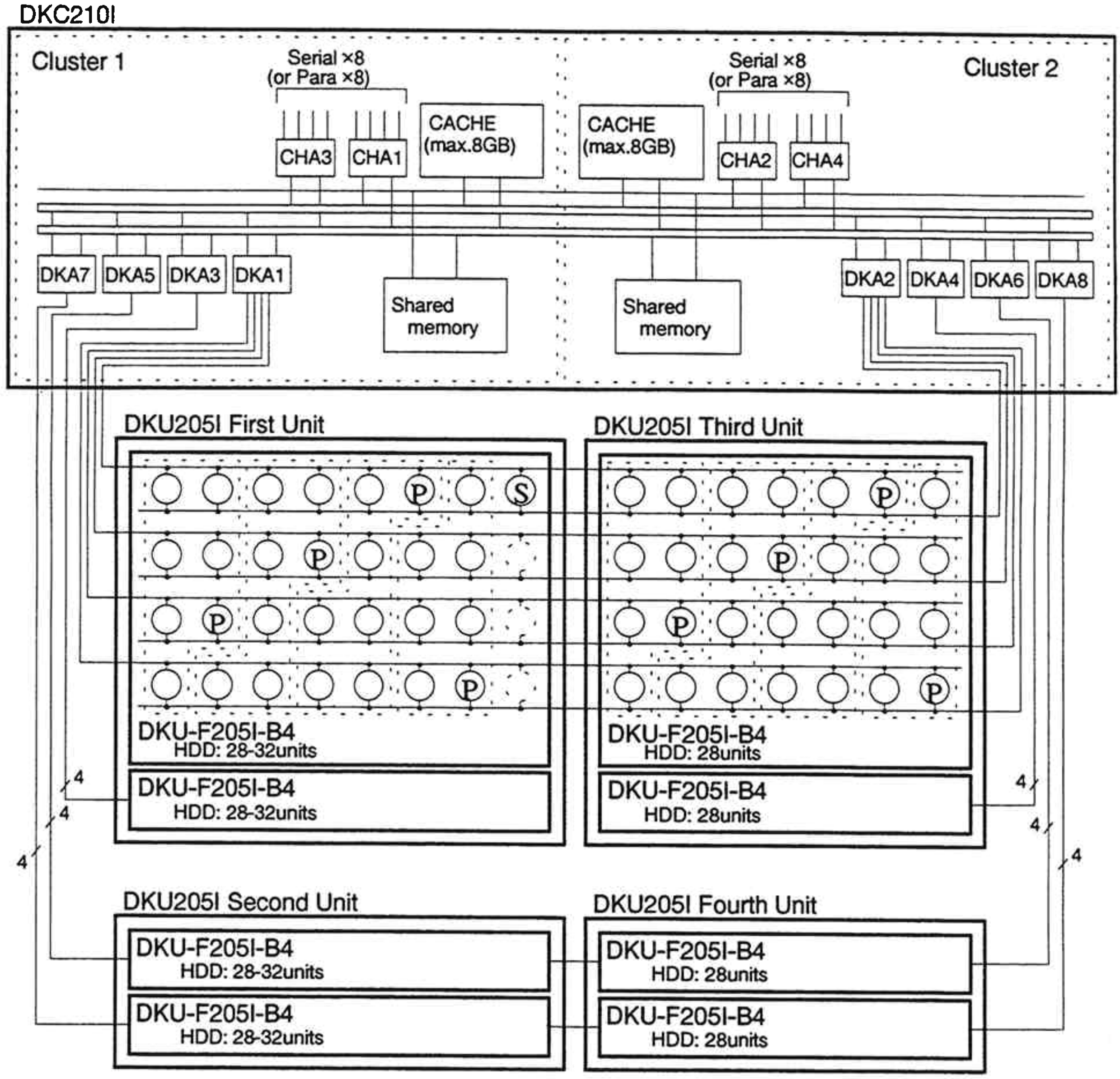


Fig. 4.3 DKC210 and DKU205 maximum Configuration

12.4.4 Data Formats

(1) Data Conversion Overview

Since the disk subsystem uses SCSI drives, data in the CKD format are converted to the FBA format on an interface before being written on the drives. The data format is shown in Fig. 4.4.1.

CKD-to-FBA conversion is carried out by the CHA. Data is stored in cache (in the DKC) in the FBA format. Consequently, the drive need not be aware of the data format when transferring to and receiving data from cache.

Each field of the CKD-format record is left-justified and the data is controlled in units of 528-byte subblocks (because data is transferred in 16-byte units). Each field is provided with data integrity code (LRC). An address integrity code (LA: logical address) is appended to the end of each subblock. A count area (C area) is always placed at the beginning of the subblock.

Four subblocks make up a single block. The first subblock of a block is provided with T information (record position information).

If a record proves not to fit in a subblock during CKD-to-FBA conversion, a field is split into the next subblock when it is recorded. If a record does not fill a subblock, the subblock is padded with 00s, from the end of the last field to the LA.

On a physical drive, data is recorded data fields in 520-byte units (physical data format). The format of the LA in the subblock in cache is shown in Fig. 4.4.1. The last 8 bytes of the LA area are padding data which is insignificant (the reason for this is because data is transferred to cache in 16 byte units). When data is transferred to a drive from cache, the last 8 bytes of each LA area are discarded and 520 bytes are transferred.

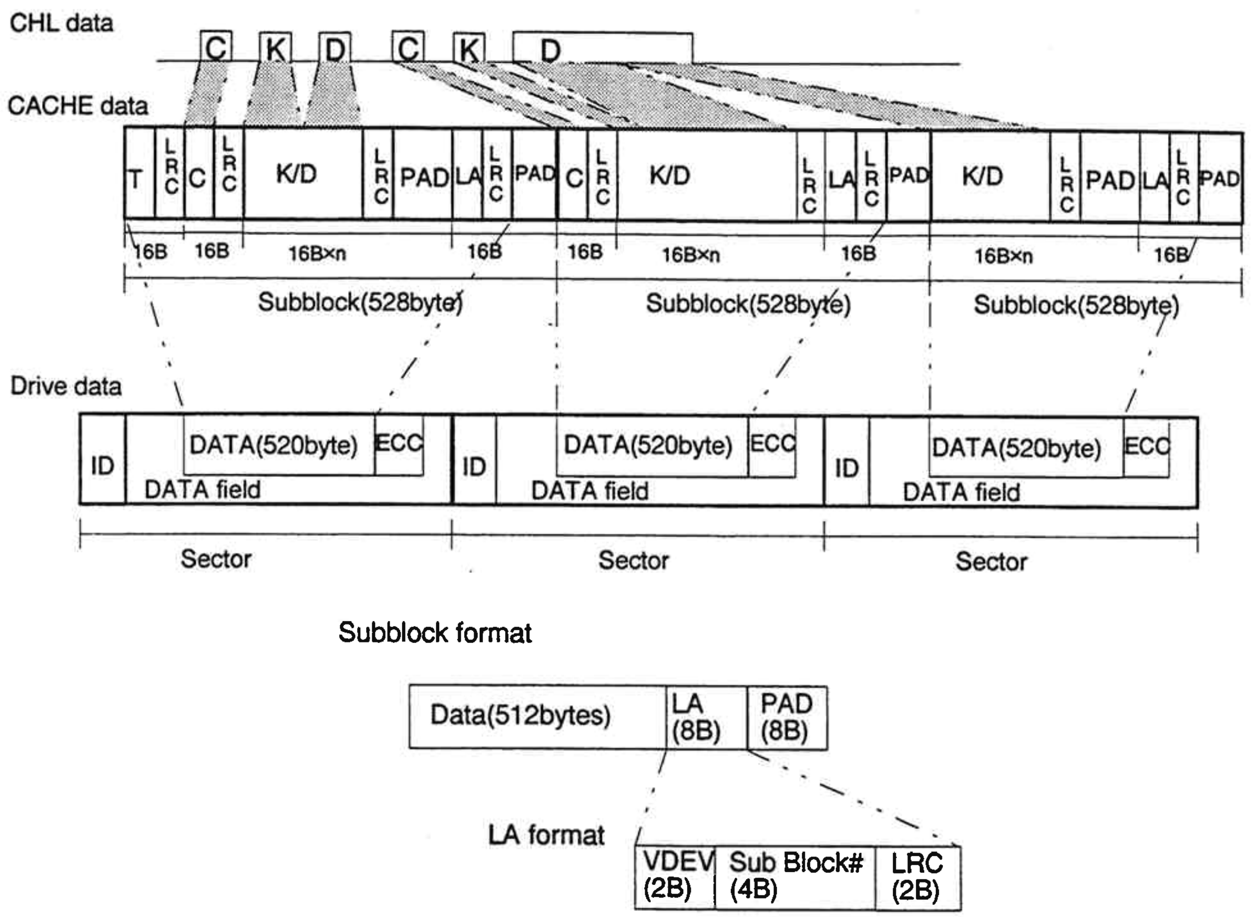


Fig. 4.4.1 Data Format

(2) Block format

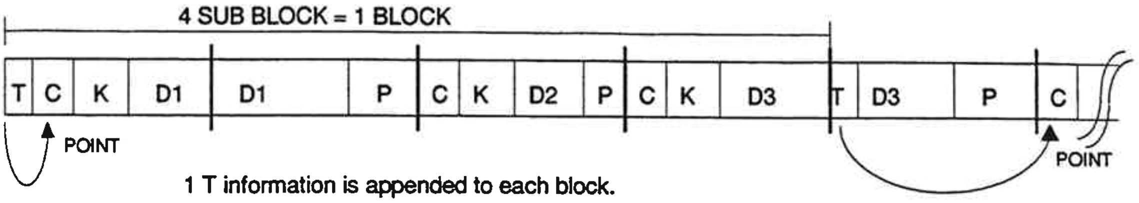


Fig. 4.4.2 Block Format

The RAID system records T information for each block of 4 subblocks as positional information that is used during record search. This unit of data is called a block.

1 block = 4 subblocks = 2 KB

The T information is 16 bytes long. However, only two bytes have meaning and the remaining 14 byte positions are padded with 0s. The reason for this is the same as that for the LA area. Unlike the LA, the insignificant bytes are also stored on the drive as are.

As seen from Fig. 4.4.2, the T information points to the closest count area in its block in the form of an SN (segment number). The drive computes the block number from the sector number with the SET SEC and searches the T information for the target block. From the T information, the drive computes the location of the closest count area and starts processing the block at the count area. This means that the information plays the role of the AM of the conventional disk storage.

(3) Data integrity provided

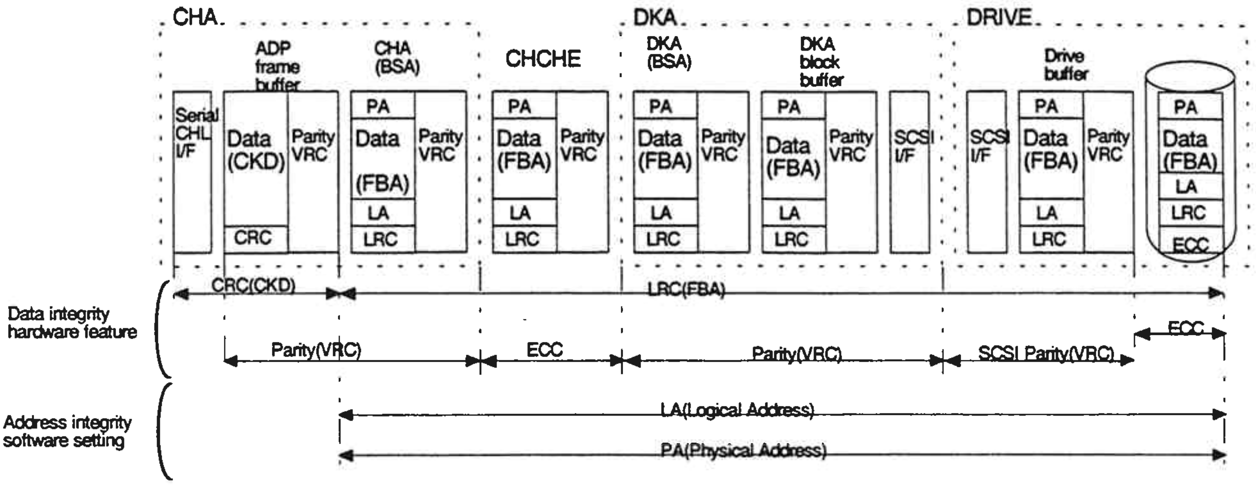


Fig. 4.4.3 Outline of Data Integrity

In the DKC210 and DKU205 system, a data integrity code is appended to the data being transferred at each component as shown in Fig. 4.4.3. Since data is striped onto two or more disk devices and the address integrity code is also appended. The data integrity codes are appended by hardware and the address integrity codes by software.

12.4.5 Cache Management

Since the DKC requires no through operation, its cache system is implemented by two memory areas called cache A and cache B so that write data can be duplexed. To prevent data loss due to power failures, cache is made non-volatile by being fully battery-backed (48 hours). This dispenses with the need for the conventional NVS.

The minimum unit of cache is the 16 KB segment. Cache is destaged in segment units. Emulation Disk type at three or four segments make up one slot. The read and write slots are always controlled in pair. Cache data is enqueued and dequeued usually in slot units. In real practice, the segments of the same slot are not always stored in a contiguous area in cache, but are stored in discreet areas. These segments are controlled suing CACHE-SLCB and CACHE-SGCB so that the segments belonging to the same slot are seemingly stored in a contiguous area in cache.

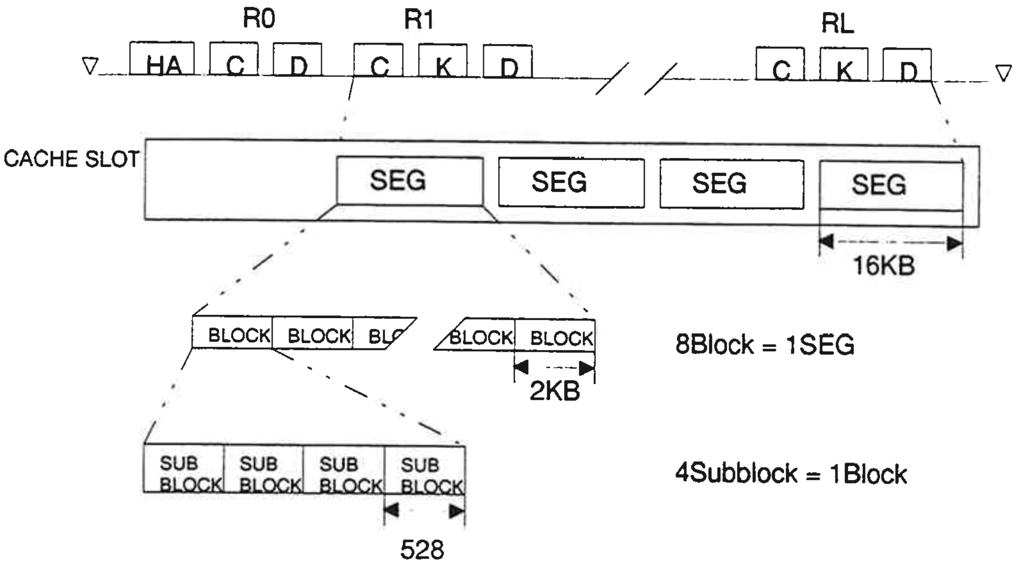


Fig. 4.5 Cache Data Structure

For increased directory search efficiency, a single virtual device (VDEV) is divided into 16-slot groups which are controlled using VDEV-GRPP and CACHE-GRPT.

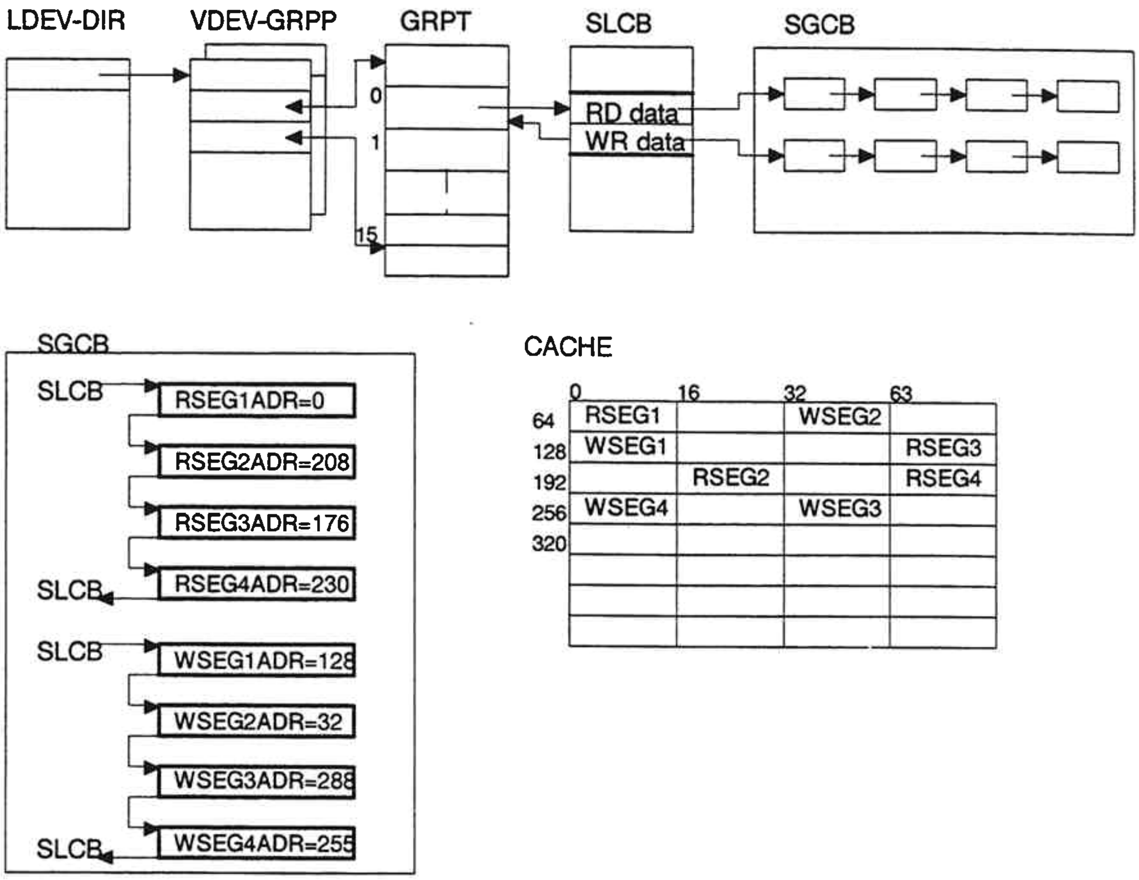
1 cache segment = 8 blocks = 32 subblocks = 16 KB
1 slot = 1 stripe = 4 segments = 64 KB

The directories VDEV-GRPP, CACHE-GRPT, CACHE-SLCB, and CACHE-SGCB are used to identify the cache hit and miss conditions. These control tables are stored in the shared memory.

In addition to cache hit and miss control, the shared memory is used to classify and control the data in cache according to its attributes. Queues are something like boxes that are used to classify data according to its attributes.

Basically, queues are controlled in slot units (some queues are controlled in segment units). Like SLCB-SGCB, queues are controlled using a queue control table so that queue data of the seemingly same attribute can be controlled as a single data group. These control tables are briefly described below.

(1) Cache control tables (directories)



	0	16	32	63
64	RSEG1		WSEG2	
128	WSEG1			RSEG3
192		RSEG2		RSEG4
256	WSEG4		WSEG3	
320				

Fig. 4.5.1 Cache Control Tables

LDEV-DIR (Logical DEV-directory):

Contains the shared memory addresses of VDEV-GRPPs for an LDEV. LDEV-DIR is located in the local memory in the CHA.

VDEV-GRPP (Virtual DEV-group Pointer):

Contains the shared memory addresses of the GRPTs associated with the group numbers in the VDEV.

GRPT (Group Table):

A table that contains the shared memory address of the SLCBs for 16 slots in the group. Slogs are grouped to facilitate slot search and to reduce the space for the directory area.

SLCB (Slot Control Block):

Contains the shared memory addresses of the starting and ending SGCBs in the slot. Two or more SGCBs are chained. The SLCB also stores lot status and points to the queue that is connected to the slot. The state transitions of clean and dirty queues occur in slot units. The processing tasks reserve and release cache areas in this unit.

SGCB (Segment Control Block):

Contains the control information about a cache segment. It contains the cache address of the segment. It is used to control the staged subblock bit map, dirty subblock bitmap, and other information. The state transitions of only free queues occur in segment units.

(2) Cache control table access method (hit/miss identification procedure)

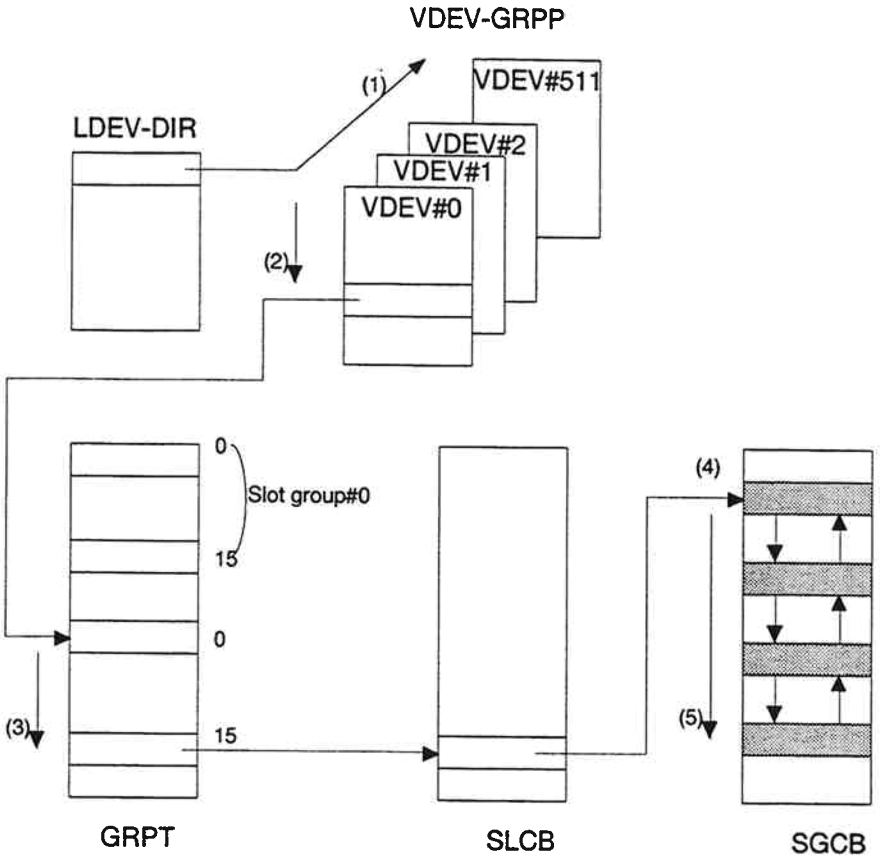


Fig. 4.5.2 Outline of Cache Control Table Access

1. The current VDEV-GRPP is referenced through the LDEV-DIR to determine the hit/miss condition of the VDEV-groups.
2. If a VDEV-group hits, CACHE-GRPT is referenced to determine the hit/miss condition of the slots.
3. If a slot hits, CACHE-SLCB is referenced to determine the hit/miss condition of the segments.
4. If a segment hits, CACHE-SGCB is referenced to access the data in cache.

If a search miss occurs during the searches from 1. through 4., the target data causes a cache miss.

Definition of VDEV number
Since the host processor recognizes addresses only by LDEV, it is unaware of the device address of the parity device. Accordingly, the RAID system is provided with a VDEV address which identifies the parity device associated with an LDEV. Since VDEVs are used to control data devices and parity devices systematically, their address can be computed using the following formulas:
Data VDEV number = LDEV number
Parity VDEV number = 256 + LDEV number

From the above formulas, the VDEV number ranges from 0 to 511.

(3) Queue structures

The DKC210 and DKU205 uses 10 types of queues to control data in cache segments according to its attributes. These queues are explained below.

- CACHE-GRPT free queue

This queue is used to control segments that are currently not used by CACHE-GRPT (free segments) on an FIFO (First In First Out) basis. When a new table is added to CACHE-GRPT, the segment that is located by the head pointer of the queue is used.

- CACHE-SLCB free queue

This queue is used to control segments that are currently not used by CACHE-SLCB (free segments) on an FIFO basis. When a new slot is added to CACHE-SLCB, the segment that is located by the head pointer of the queue is used.

- CACHE-SGCB free queue

This queue is used to control segments that are currently not used by CACHE-SGCB (free segments) on an FIFO basis. When a new segment is added to CACHE-SGCB, the segment that is located by the head pointer of the queue is used.

- Clean queue

This queue is used to control the segments that are reflected on the drive on an LRU basis.

- Bind queue (M-series only)

This queue is defined when the bind mode is specified and used to control the segments of the bind attribute on an LRU basis.

- Error queue

This queue controls the segments that are no longer reflected on the drive due to some error (pinned data) on an LRU basis.

- Parity in-creation queue

This queue controls the slots (segments) that are creating parity on an LRU basis.

- DFW queue (host dirty queue)

This queue controls the segments that are not reflected on the drive in the DFW mode on an LRU basis. There are 16 DFW qeueues.

- CFW queue (host dirty queue)

This queue controls the segments that are not reflected on the drive in the CFW mode on an LRU basis.

- PDEV queue (physical dirty queue)

This queue controls the data (segments) that are not reflected on the drive and that occur after parity is generated. Data is staged from this queue onto the physical DEV. There are 32 PDEV queues per physical DEV.

The control table for these queues is located in the shared memory and points to the head and tail segments of the queues.

(4) Queue state transitions

Figure 4.5.4 shows the state transitions of the queues used in. A brief description of the queue state transitions follows.

- State transition from a free queue
When a read miss occurs, the pertinent segment is staged and enqueued to a clean queue. When a write miss occurs, the pertinent segment is temporarily staged and enqueued to a host dirty queue.

- State transition from a clean queue
When a write hit occurs, the segment is enqueued to a host dirty queue. Transition from clean to free queues is performed on an LRU basis.

- State transition from a host dirty queue
The host dirty queue contains data that reflects no parity. When parity generation is started, a state transition occurs to the parity in-creation queue.

- State transition from the parity in-creation queue
The parity in-creation queue contains parity in-creation data. When parity generation is completed, a transition to a physical dirty queue occurs.

- State transition from a physical dirty queue
When a write hit occurs in the data segment that is enqueued in a physical dirty queue, the segment is enqueued into the host dirty queue again. When destaging of the data segment is completed, the segment is enqueued into a queue (destaging of data segments occur asynchronously on an LRU basis).

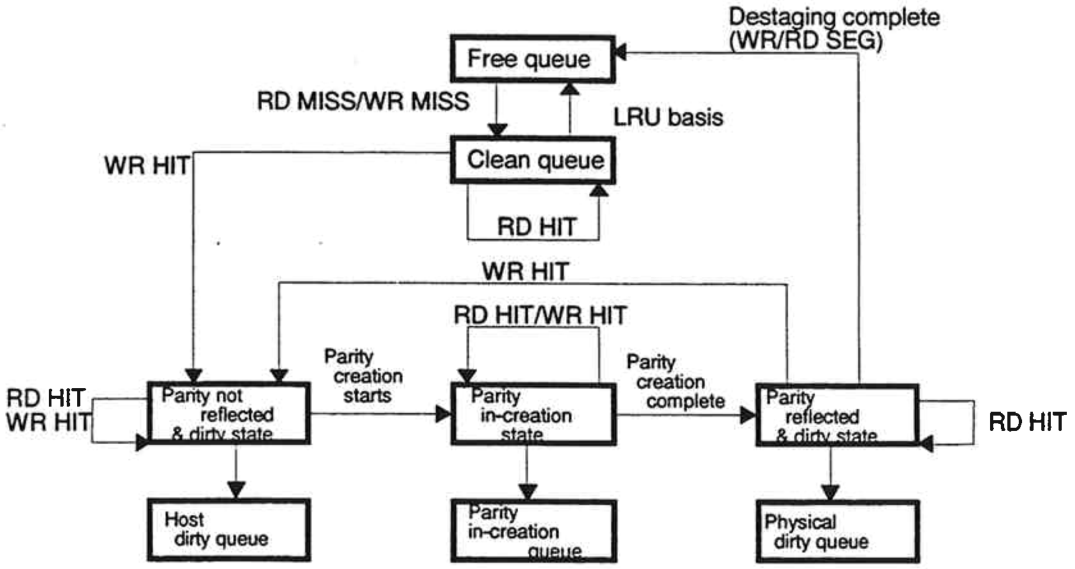
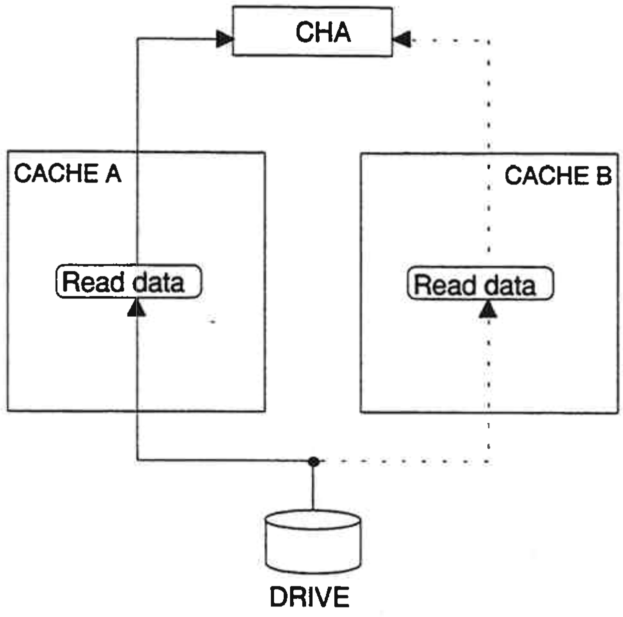


Fig. 4.5.4 Queue Segment State Transition Diagram

(5) Cache usage in the read mode



The cache area to be used for destaging read data is determined depending on whether the result of evaluating the following expression is odd or even:

(CYL# x 15 + HD#) / 16

The read data is destaged into area A if the result is even and into area B if the result is odd.

Fig. 4.5.5 Cache Usage in the Read Mode

Read data is not duplexed and its destaging cache area is determined by the formula shown in Fig. 4.5.5. Staging is performed not only on the segments containing the pertinent block but also on the subsequent segments up to the end of track (for increased hit ratio). Consequently, one track equivalence of data is prefetched starting at the target block. This formula is introduced so that the cache activity ratios for areas A and B are even. The staged cache area is called the cache area and the other area NVS area.

(6) Cache usage in the write mode

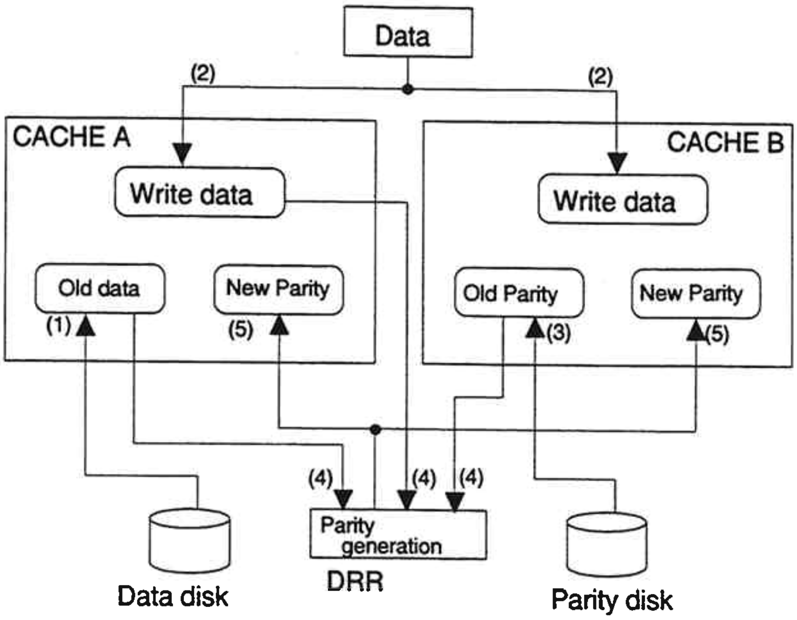


Fig. 4.5.6 Cache Usage in the Write Mode

This system handles write data (new data) and read data (old data) in separate segments as shown in Fig. 4.5.6 (not overwritten as in the conventional systems), whereby compensating for the write penalty.

(1) If the write data in question causes a cache miss, the data from the block containing the target record up to the end of the track is staged into a read data slot.
(2) In parallel with step (1), the write data is transferred when the block in question is established in the read data slot.
(3) The parity data for the block in question is checked for a hit or miss condition and, if a cache miss condition is detected, the old parity is staged into a read parity slot.
(4) When all data necessary for generating new parity is established, it is transferred to the DRR circuit in the DKA.
(5) When the new parity is completed, the DRR transfers it into the write parity slots for cache A and cache B (the new parity is handled in the same manner as the write data).

The reason for writing the write data into both cache areas is that data will be lost if a cache error occurs when it is not yet written on the disk.

Although two cache areas are used as explained above, the read data (including parity) is destaged into either cache A or cache B simply by duplexing only the write data (including parity) (in the same manner as in the read mode).

(7) Pseudo Through

The DKC80/90 systems write data directly onto disk storage in the form of cache through, without performing a DFW, when a cache error occurs. In this system, cache must always be passed, which fact disables the through operation. Consequently, the write data is duplexed, and a pseudo through operation is performed; that is, when one cache subsystem goes down, the end of processing status is not reported until the data write in the other cache subsystem is completed. This process is called pseudo through.

The control information necessary for controlling cache is stored in the shared memory.

(8) Shared memory

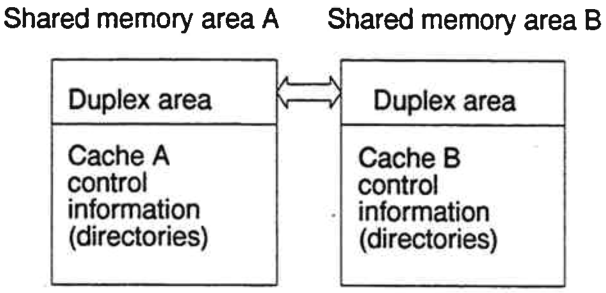

Shared memory area A
Shared memory area B
Duplex area
Duplex area
Cache A control information (directories)
Cache B control information (directories)

Fig. 4.5.7 Outline of Shared Memory

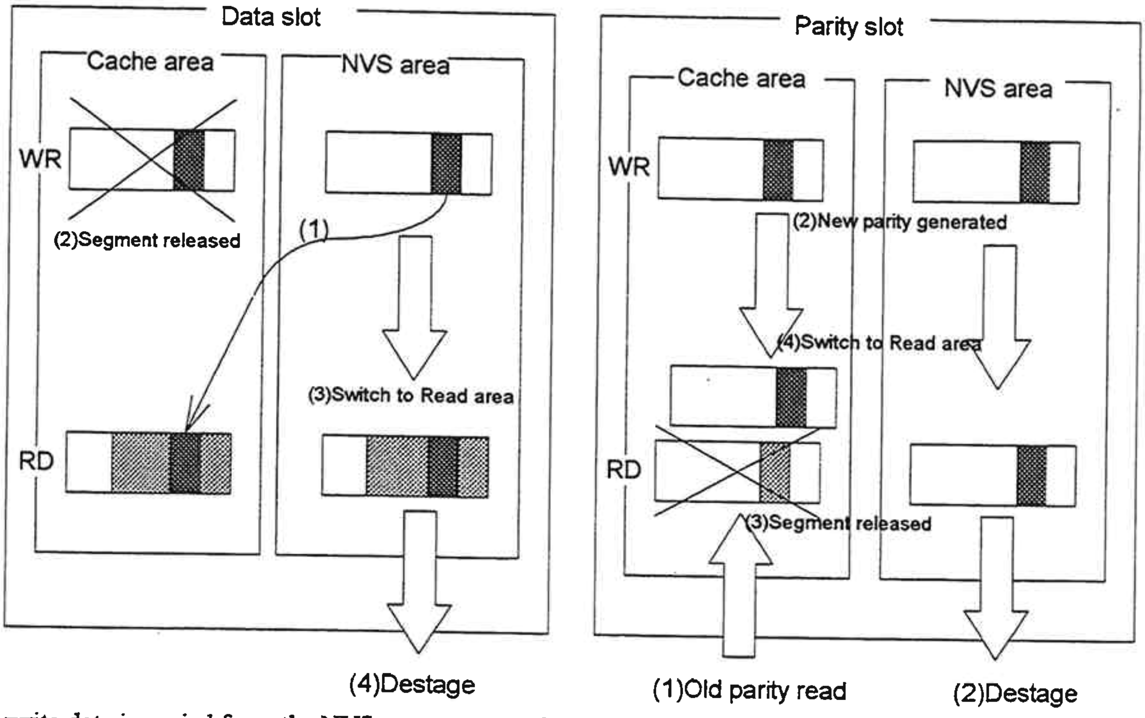
This system has two areas of cache memory, shown in Fig. 4.5.7, as it has two areas of cache memory. One part of its internal data is fully duplexed (this serves as the role of the conventional ECM). The other part of the shared memory area contains the control information about the corresponding cache area (shared memory area A for cache A and shared memory area B for cache B). If an error occurs on one side of shared memory (A or B), the corresponding cache area becomes inoperative (equivalent to a cache error).

Like cache, shared memory is made non-volatile (approximately 96 hours) to prevent data loss in case of power failures.

12.4.6 Destaging Operations

(1) Cache management in the destage mode (RAID5)

Destaging onto a drive is deferred until parity generation is completed. Data and parity slot transitions in the destage mode occur as shown in Fig. 4.6.1



① The write data is copied from the NVS area into the read area.

② The write segment in the cache area is released.

③ Simultaneously, the segment in the NVS area is switched from write to read segment.

④ Destaging.

⑤ The read segment in the NVS area is released.

①Parity generation correction read (old parity) occurs.

② New parity is generated.

③ The old parity in the read segment is released.

④ The segments in the cache and NVS areas are switched from write to read segment.

⑤ Destaging.

⑥ The read segment in the NVS area is released.

Fig. 4.6.1 Cache Operation in the Destage Mode

Write data is stored in write segments before parity is generated but stored in read segments after parity is generated. When drive data is stored, therefore, the data from the read segment is transferred.

(2) Cache management in the destage mode (RAID1)

Data slot read segment is copied to parity slot then destage to primary/secondary drive synchronously.

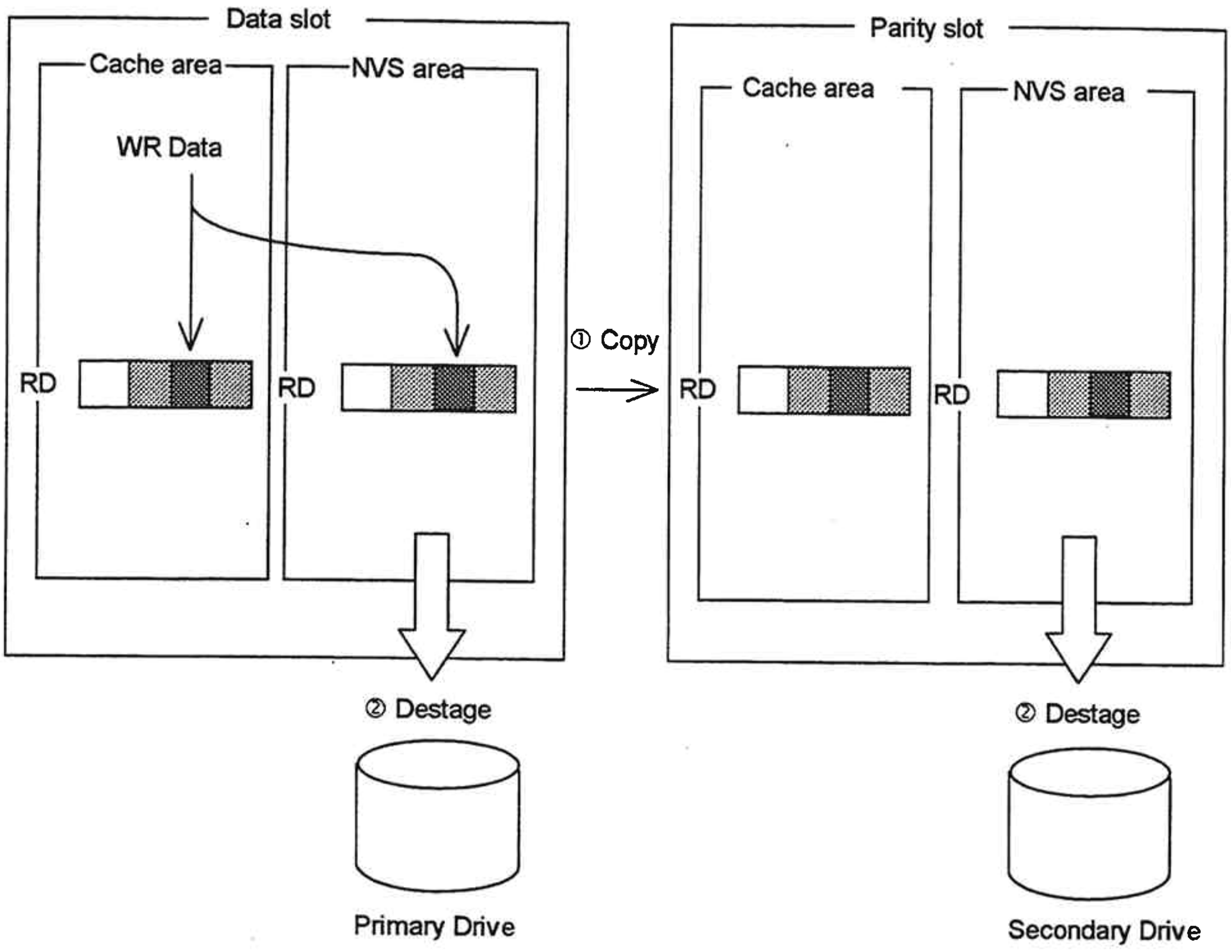


Fig. 4.6.2 RAID1 synchronous destage

① Copy the data slot read segment to the parity slot.

② Destage.

③ The data read segment in the NVS area of the data slot and the parity slot are released.

Data slot is destage to primary/secondary drive.

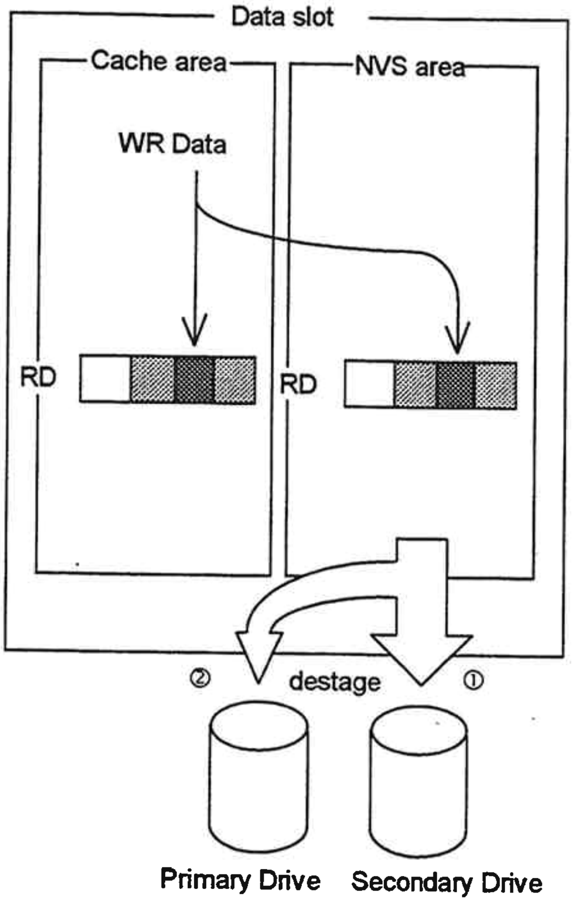


Fig. 4.6.3 RAID1 asynchronous destage

① Destage to secondary drive.
② Destage to primary drive.
③ The data read segment in the NVS area is released.

(3) Blocked data write

The purpose of blocked data write is to reduce the number of accesses to the drive during destaging, whereby increasing the subsystem performance. There are three modes of blocked data write: single-stripe blocking, multiple-stripe blocking, and drive blocking. These modes are briefly explained below.

- Single-stripe blocking

Two or more dirty segments in a stripe are combined into a single dirty data block. Contiguous dirty blocks are placed in a single area. If an unloaded block exists between dirty blocks, the system destages the dirty blocks separately at the unloaded block. If a clean block exists between dirty blocks, the system destages the blocks including the clean block.

- Multiple-stripe blocking

The sequence of stripes in a parity group are blocked to reduce the number of write penalties. This mode is useful for sequential data transfer.

- Drive blocking

In the drive blocking mode, blocks to be destaged are written in a block with a single drive command if they are contiguous when viewed from a physical drive to shorten the drive's latency time.

The single- and multiple-stripe blocking modes are also called in-cache blocking modes. The DMP determines which mode to use. The drive blocking mode is identified by the DSP.

12.4.7 Operations Performed when Drive Errors Occur

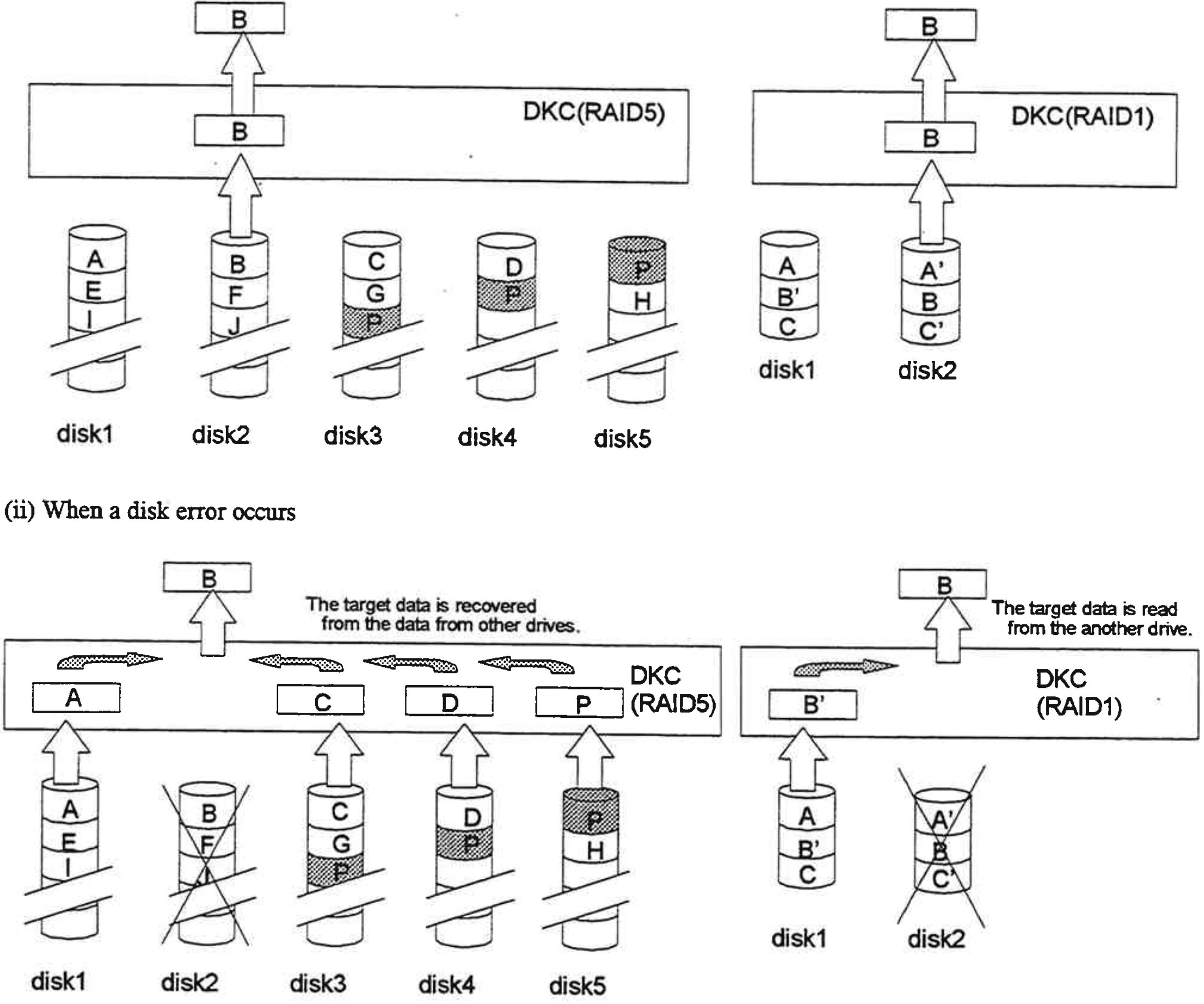
(1) I/O operations performed when drive errors occur

This system can recover target data using parity data and data stored on normal disk storage even when it cannot read data due to errors occurring on physical drives. This feature ensures non-disruptive processing of applications in case of drive errors. This system can also continue processing for the same reason in case errors occur on physical drives while processing write requests.

Figure 4.7.1 shows the outline of data read processing in case a drive error occurs.

Request for reading data B

(i) Normal time



(ii) When a disk error occurs

A,B,C • • • ; Data (A=A', B=B', C=C')

P ; Parity data

Fig. 4.7.1 Outline of Data Read Processing

(2) Data integrity feature and drive errors

This system uses spare disk drives and reconfigures any drives that are blocked due to errors or drives whose error count exceeds a specified limit value using spare disks.

Since this processing is executed on the host in the background, this system can continue to accept I/O requests. The data saved on spare disks are copied into the original location after the error drives are replaced with new ones.

1. Dynamic sparing

This system keeps track of the number of errors that occurred, for each drive, when it executes normal read or write processing. If the number of errors occurring on a certain drive exceeds a predetermined value, this system considers that the drive is likely to cause unrecoverable errors and automatically copies data from that drive to a spare disk. This function is called dynamic sparing.In RAID1 method, this system is same as RAID5 dynamic sparing.

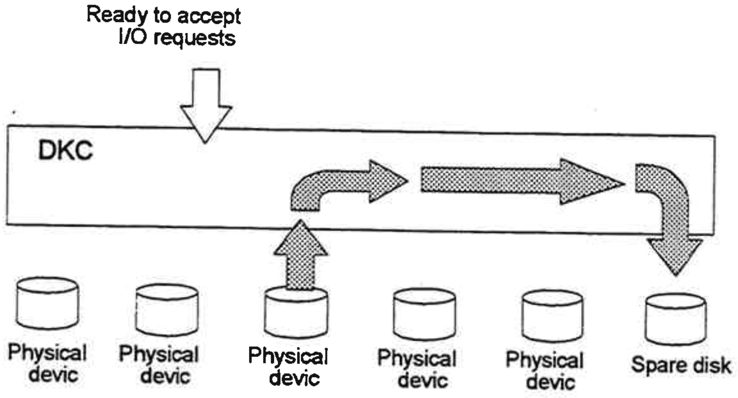


Fig. 4.7.2 Outline of the Dynamic Sparing Function

2. Correction copy

When this system cannot read or write data from or to a drive due to an error occurring on that drive, it regenerates the original data for that drive using data from the other drives and the parity data, and copies it onto a spare disk. In RAID 0 method, this system does not execute Correction Copy. In RAID1 method, this system copies data from the another drive to a spare disk.

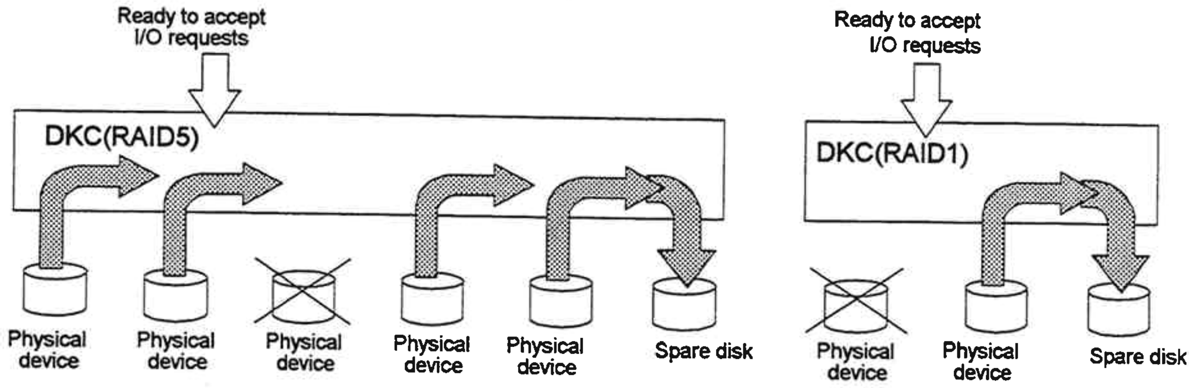


Fig. 4.7.3 Outline of the Correction Copy Function

12.5. Power-on Sequences

12.5.1 IMPL Sequence

The IMPL sequence, which is executed when power is turned on, is executed by the following four modules:

(1) Boot loader

The boot loader performs the minimum necessary amount of initializations after a ROM boot. Subsequently, the boot loader expands the local memory loader from flash memory into local memory and transfers control to the local memory loader.

(2) Local memory loader

The local memory loader loads the Vx-works load modules into local memory. Subsequently, the local memory loader transfers control to Vx-works.

(3) Vx-works

Vx-works is a root task that initializes the tables in local memory that are used for intertask communications. Vx-works also tests the hardware resources.

(4) DKC task

When the DKC task is created, it executes initialization routines. It initializes the most part of the environment it uses. When the environment is established so that the DKC task can start scanning, it notifies the SVP of a power event log. Subsequently, the DKC task turns on the power to the physical drives and, when the logical drives get ready, notifies the host processor of an NRTR.

The control flow of IMPL processing is shown in Fig. 5.1.

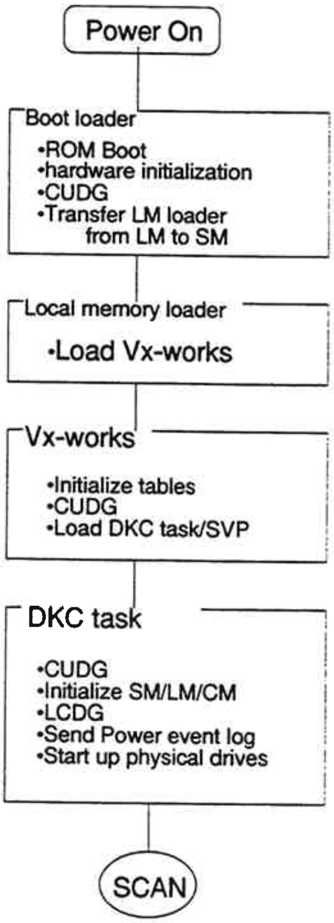


Fig. 5.1 IMPL Sequence

12.5.2 Drive Power-on Sequence

An overcurrent condition will occur if two or more drives are started at the same time. To preclude the overcurrent condition, DKUs are started at the power supply level, one at a time, at approximately 10 second intervals (32 DKUs at a time in the maximum configuration (256 LDEVs)).

When the logical devices get ready as the result of the startup of the physical drives, the host processor is notified to that effect.

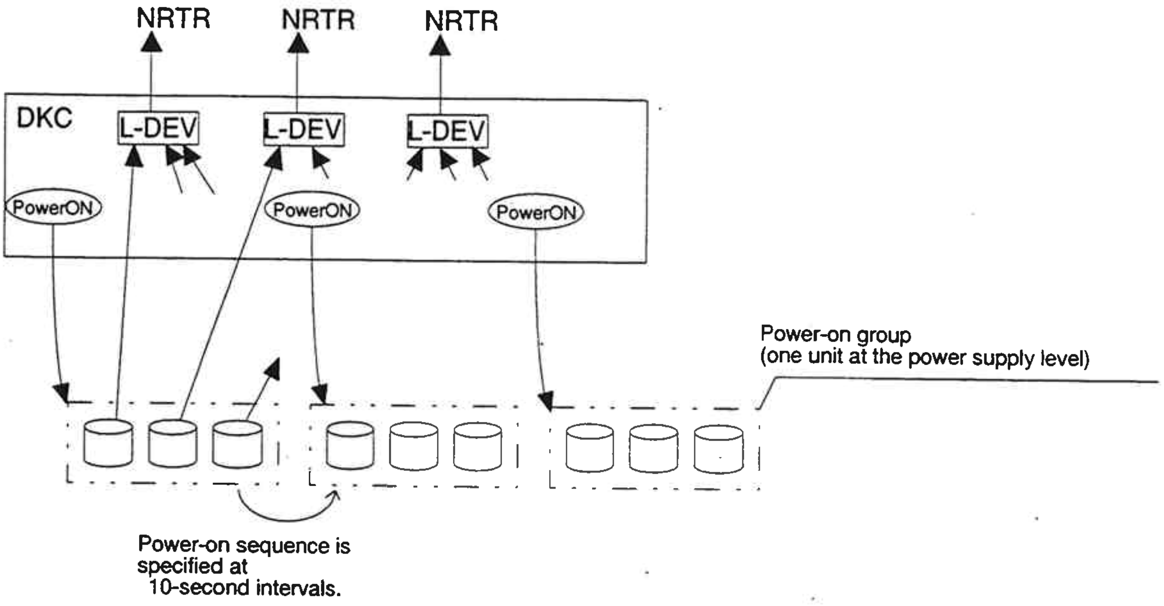


Fig. 5.2 Drive Power-on Sequence

12.5.3 Planned Stop

When a power-off is specified by a maintenance personnel, this subsystem checks for termination of tasks that are blocked or running on all logical devices. When all the tasks are terminated, this subsystem disables the CHL and executes emergency destaging. If a track for which destaging fails (pinned track) occurs, this subsystem stores the pin information in shared memory. Subsequently, this subsystem saves the pin information, which is used as hand-over information, in flash memory, sends Power Event Log to the SVP, and notifies the hardware of the grant to turn off the power.

The hardware turns off main power when power-off grants for all processors are presented.

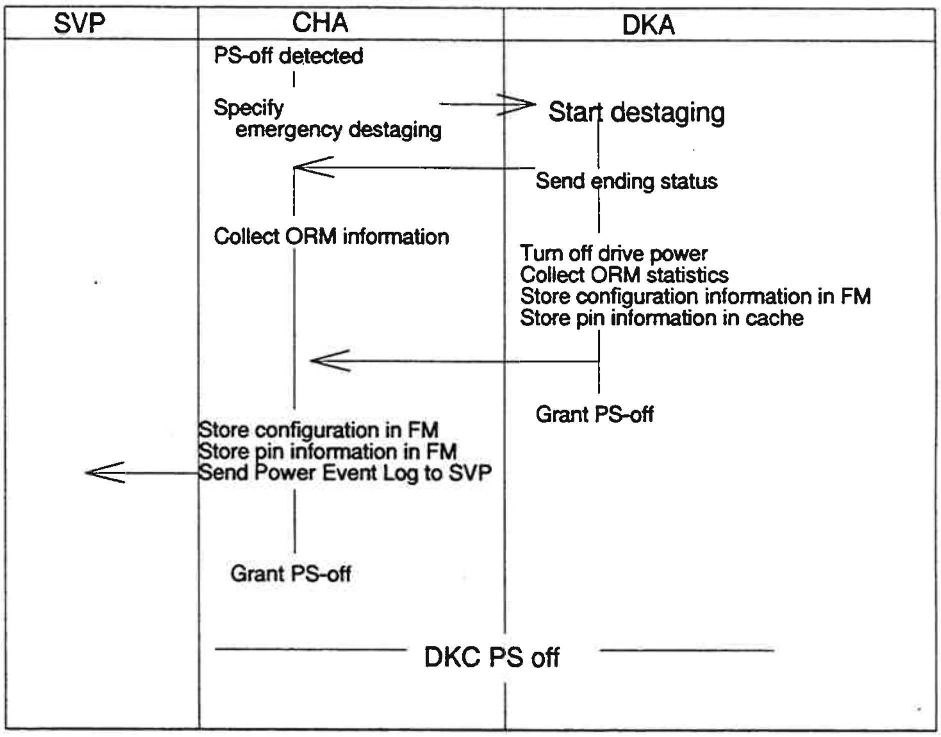


Fig. 5.3 Planned Stop Sequence

12.6. Appendixes

12.6.1 PDEV No.-C#/R# Matrixes

Table 6.1 PDEV No.-C#/R# Matrixes (1/6)

HDU No	PDEV No.	C# / R#	RAID Group#			
			RAID5		RAID1	RAID0
			6D+1P	3D+1P		
HDU-R10	HDU-R100	00 / 01	01 - 01	01 - 02	01 - 03	01 - 05
	HDU-R101	00 / 03	01 - 02	01 - 04	01 - 07	01 - 12
	HDU-R102	00 / 05	01 - 03	01 - 06	01 - 11	01 - 19
	HDU-R103	00 / 07	Spare	Spare	Spare	01 - 29
	HDU-R104	00 / 06	01 - 04	01 - 07	01 - 13	01 - 22
	HDU-R105	00 / 04	01 - 03	01 - 05	01 - 09	01 - 15
	HDU-R106	00 / 02	01 - 02	01 - 03	01 - 05	01 - 08
	HDU-R107	00 / 00	01 - 01	01 - 01	01 - 01	01 - 01
HDU-R11	HDU-R110	01 / 01	01 - 01	01 - 02	01 - 03	01 - 06
	HDU-R111	01 / 03	01 - 02	01 - 04	01 - 07	01 - 13
	HDU-R112	01 / 05	Spare	01 - 06	01 - 11	01 - 30
	HDU-R113	01 / 07	01 - 04	Spare	Spare	01 - 26
	HDU-R114	01 / 06	01 - 04	01 - 07	01 - 13	01 - 23
	HDU-R115	01 / 04	01 - 03	01 - 05	01 - 09	01 - 16
	HDU-R116	01 / 02	01 - 02	01 - 03	01 - 05	01 - 19
	HDU-R117	01 / 00	01 - 01	01 - 01	01 - 01	01 - 02
HDU-R12	HDU-R120	02 / 01	01 - 01	01 - 02	01 - 04	01 - 07
	HDU-R121	02 / 03	Spare	01 - 04	01 - 08	01 - 31
	HDU-R122	02 / 05	01 - 03	01 - 06	01 - 12	01 - 20
	HDU-R123	02 / 07	01 - 04	Spare	Spare	01 - 27
	HDU-R124	02 / 06	01 - 04	01 - 07	01 - 14	01 - 24
	HDU-R125	02 / 04	01 - 03	01 - 05	01 - 10	01 - 17
	HDU-R126	02 / 02	01 - 02	01 - 03	01 - 06	01 - 10
	HDU-R127	02 / 00	01 - 01	01 - 01	01 - 02	01 - 03
HDU-R13	HDU-R130	03 / 01	Spare	01 - 02	01 - 04	01 - 32
	HDU-R131	03 / 03	01 - 02	01 - 04	01 - 08	01 - 14
	HDU-R132	03 / 05	01 - 03	01 - 06	01 - 12	01 - 21
	HDU-R133	03 / 07	01 - 04	Spare	Spare	01 - 28
	HDU-R134	03 / 06	01 - 04	01 - 07	01 - 14	01 - 25
	HDU-R135	03 / 04	01 - 03	01 - 05	01 - 10	01 - 18
	HDU-R136	03 / 02	01 - 02	01 - 03	01 - 06	01 - 11
	HDU-R137	03 / 00	01 - 01	01 - 01	01 - 02	01 - 04
HDU-R14	HDU-R140	04 / 01	02 - 01	02 - 02	02 - 03	02 - 05
	HDU-R141	04 / 03	02 - 02	02 - 04	02 - 07	02 - 12
	HDU-R142	04 / 05	02 - 03	02 - 06	02 - 11	02 - 19
	HDU-R143	04 / 07	Spare	Spare	Spare	02 - 29
	HDU-R144	04 / 06	02 - 04	02 - 07	02 - 13	02 - 22
	HDU-R145	04 / 04	02 - 03	02 - 05	02 - 09	02 - 15
	HDU-R146	04 / 02	02 - 02	02 - 03	02 - 05	02 - 08
	HDU-R147	04 / 00	02 - 01	02 - 01	02 - 01	02 - 01
HDU-R15	HDU-R150	05 / 01	02 - 01	02 - 02	02 - 03	02 - 06
	HDU-R151	05 / 03	02 - 02	02 - 04	02 - 07	02 - 13
	HDU-R152	05 / 05	Spare	02 - 06	02 - 11	02 - 30
	HDU-R153	05 / 07	02 - 04	Spare	Spare	02 - 26
	HDU-R154	05 / 06	02 - 04	02 - 07	02 - 13	02 - 23
	HDU-R155	05 / 04	02 - 03	02 - 05	02 - 09	02 - 16
	HDU-R156	05 / 02	02 - 02	02 - 03	02 - 05	02 - 09
	HDU-R157	05 / 00	02 - 01	02 - 01	02 - 01	02 - 02

Table 6.1 PDEV No.-C#/R# Matrixes (2/6)

HDU No	P DEV No.	C# / R#	RAID Group#			
			RAID5		RAID1	RAID0
			6D+1P	3D+1P		
HDU-R16	HDU-R160	06 / 01	02 - 01	02 - 02	02 - 04	02 - 07
	HDU-R161	06 / 03	Spare	02 - 04	02 - 08	02 - 31
	HDU-R162	06 / 05	02 - 03	02 - 06	02 - 12	02 - 20
	HDU-R163	06 / 07	02 - 04	Spare	Spare	02 - 27
	HDU-R164	06 / 06	02 - 04	02 - 07	02 - 14	02 - 24
	HDU-R165	06 / 04	02 - 03	02 - 05	02 - 10	02 - 17
	HDU-R166	06 / 02	02 - 02	02 - 03	02 - 06	02 - 10
	HDU-R167	06 / 00	02 - 01	02 - 01	02 - 02	02 - 03
HDU-R17	HDU-R170	07 / 01	Spare	02 - 02	02 - 04	02 - 32
	HDU-R171	07 / 03	02 - 02	02 - 04	02 - 08	02 - 14
	HDU-R172	07 / 05	02 - 03	02 - 06	02 - 12	02 - 21
	HDU-R173	07 / 07	02 - 04	Spare	Spare	02 - 28
	HDU-R174	07 / 06	02 - 04	02 - 07	02 - 14	02 - 25
	HDU-R175	07 / 04	02 - 03	02 - 05	02 - 10	02 - 18
	HDU-R176	07 / 02	02 - 02	02 - 03	02 - 06	02 - 11
	HDU-R177	07 / 00	02 - 01	02 - 01	02 - 02	02 - 04
HDU-L10	HDU-L100	08 / 01	03 - 01	03 - 02	03 - 03	03 - 05
	HDU-L101	08 / 03	03 - 02	03 - 04	03 - 07	03 - 12
	HDU-L102	08 / 05	03 - 03	03 - 06	03 - 11	03 - 19
	HDU-L103	08 / 07	Spare	Spare	Spare	03 - 29
	HDU-L104	08 / 06	03 - 04	03 - 07	03 - 13	03 - 22
	HDU-L105	08 / 04	03 - 03	03 - 05	03 - 09	03 - 15
	HDU-L106	08 / 02	03 - 02	03 - 03	03 - 05	03 - 08
	HDU-L107	08 / 00	03 - 01	03 - 01	03 - 01	03 - 01
HDU-L11	HDU-L110	09 / 01	03 - 01	03 - 02	03 - 03	03 - 06
	HDU-L111	09 / 03	03 - 02	03 - 04	03 - 07	03 - 13
	HDU-L112	09 / 05	Spare	03 - 06	03 - 11	03 - 30
	HDU-L113	09 / 07	03 - 04	Spare	Spare	03 - 26
	HDU-L114	09 / 06	03 - 04	03 - 07	03 - 13	03 - 23
	HDU-L115	09 / 04	03 - 03	03 - 05	03 - 09	03 - 16
	HDU-L116	09 / 02	03 - 02	03 - 03	03 - 05	03 - 09
	HDU-L117	09 / 00	03 - 01	03 - 01	03 - 01	03 - 02
HDU-L12	HDU-L120	0A / 01	03 - 01	03 - 02	03 - 04	03 - 07
	HDU-L121	0A / 03	Spare	03 - 04	03 - 08	03 - 31
	HDU-L122	0A / 05	03 - 03	03 - 06	03 - 12	03 - 20
	HDU-L123	0A / 07	03 - 04	Spare	Spare	03 - 27
	HDU-L124	0A / 06	03 - 04	03 - 07	03 - 14	03 - 24
	HDU-L125	0A / 04	03 - 03	03 - 05	03 - 10	03 - 17
	HDU-L126	0A / 02	03 - 02	03 - 03	03 - 06	03 - 10
	HDU-L127	0A / 00	03 - 01	03 - 01	03 - 02	03 - 03
HDU-L13	HDU-L130	0B / 01	Spare	03 - 02	03 - 04	03 - 32
	HDU-L131	0B / 03	03 - 02	03 - 04	03 - 08	03 - 14
	HDU-L132	0B / 05	03 - 03	03 - 06	03 - 12	03 - 21
	HDU-L133	0B / 07	03 - 04	Spare	Spare	03 - 28
	HDU-L134	0B / 06	03 - 04	03 - 07	03 - 14	03 - 25
	HDU-L135	0B / 04	03 - 03	03 - 05	03 - 10	03 - 18
	HDU-L136	0B / 02	03 - 02	03 - 03	03 - 06	03 - 11
	HDU-L137	0B / 00	03 - 01	03 - 01	03 - 02	03 - 04

Table 6.1 PDEV No.-C#/R# Matrixes (3/6)

HDU No	P DEV No.	C# / R#	RAID Group#			
			RAID5		RAID1	RAID0
			6D+1P	3D+1P		
HDU-L14	HDU-L140	0C / 01	04 - 01	04 - 02	04 - 03	04 - 05
	HDU-L141	0C / 03	04 - 02	04 - 04	04 - 07	04 - 12
	HDU-L142	0C / 05	04 - 03	04 - 06	04 - 11	04 - 19
	HDU-L143	0C / 07	Spare	Spare	Spare	04 - 29
	HDU-L144	0C / 06	04 - 04	04 - 07	04 -- 13	04 - 22
	HDU-L145	0C / 04	04 - 03	04 - 05	04 - 09	04 - 15
	HDU-L146	0C / 02	04 - 02	04 - 03	04 - 05	04 - 08
	HDU-L147	0C / 00	04 - 01	04 - 01	04 - 01	04 - 01
HDU-L15	HDU-L150	0D / 01	04 - 01	04 - 02	04 - 03	04 - 06
	HDU-L151	0D / 03	04 - 02	04 - 04	04 - 07	04 - 13
	HDU-L152	0D / 05	Spare	04 - 06	04 - 11	04 - 30
	HDU-L153	0D / 07	04 - 04	Spare	Spare	04 - 26
	HDU-L154	0D / 06	04 - 04	04 - 07	04 - 13	04 - 23
	HDU-L155	0D / 04	04 - 03	04 - 05	04 - 09	04 - 16
	HDU-L156	0D / 02	04 - 02	04 - 03	04 - 05	04 - 09
	HDU-L157	0D / 00	04 - 01	04 - 01	04 - 01	04 - 02
HDU-L16	HDU-L160	0E / 01	04 - 01	04 - 02	04 - 04	04 - 07
	HDU-L161	0E / 03	Spare	04 - 04	04 - 08	04 - 31
	HDU-L162	0E / 05	04 - 03	04 - 06	04 - 12	04 - 20
	HDU-L163	0E / 07	04 - 04	Spare	Spare	04 - 27
	HDU-L164	0E / 06	04 - 04	04 - 07	04 - 14	04 - 24
	HDU-L165	0E / 04	04 - 03	04 - 05	04 - 10	04 - 17
	HDU-L166	0E / 02	04 - 02	04 - 03	04 - 06	04 - 10
	HDU-L167	0E / 00	04 - 01	04 - 01	04 - 02	04 - 03
HDU-L17	HDU-L170	0F / 01	Spare	04 - 02	04 - 04	04 - 32
	HDU-L171	0F / 03	04 - 02	04 - 04	04 - 08	04 - 14
	HDU-L172	0F / 05	04 - 03	04 - 06	04 - 12	04 - 21
	HDU-L173	0F / 07	04 - 04	Spare	Spare	04 - 28
	HDU-L174	0F / 06	04 - 04	04 - 07	04 - 14	04 - 25
	HDU-L175	0F / 04	04 - 03	04 - 05	04 - 10	04 - 18
	HDU-L176	0F / 02	04 - 02	04 - 03	04 - 06	04 - 11
	HDU-L177	0F / 00	04 - 01	04 - 01	04 - 02	04 - 04
HDU-R20	HDU-R200	00 / 09	05 - 01	05 - 01	05 - 01	05 - 05
	HDU-R201	00 / 0B	05 - 02	05 - 03	05 - 05	05 - 12
	HDU-R202	00 / 0D	05 - 03	-	05 - 09	05 - 19
	HDU-R203	00 / 08	05 - 01	-	05 - 13	05 - 01
	HDU-R204	00 / 0E	05 - 04	-	05 - 11	05 - 22
	HDU-R205	00 / 0C	05 - 03	05 - 04	05 - 07	05 - 15
	HDU-R206	00 / 0A	05 - 02	05 - 02	05 - 03	05 - 08
HDU-R21	HDU-R210	01 / 09	05 - 01	05 - 01	05 - 01	05 - 06
	HDU-R211	01 / 0B	05 - 02	05 - 03	05 - 05	05 - 13
	HDU-R212	01 / 08	05 - 01	-	05 - 09	05 - 02
	HDU-R213	01 / 0E	05 - 04	-	05 - 13	05 - 26
	HDU-R214	01 / 0D	05 - 04	-	05 - 11	05 - 23
	HDU-R215	01 / 0C	05 - 03	05 - 04	05 - 07	05 - 16
	HDU-R216	01 / 0A	05 - 02	05 - 02	05 - 03	05 - 09

Table 6.1 PDEV No.-C#/R# Matrixes (4/6)

HDU No	P DEV No.	C# / R#	RAID Group#			
			RAID5		RAID1	RAID0
			6D+1P	3D+1P		
HDU-R22	HDU-R220	02 / 09	05 - 01	05 - 01	05 - 02	05 - 07
	HDU-R221	02 / 08	05 - 01	05 - 03	05 - 06	05 - 03
	HDU-R222	02 / 0C	05 - 03	-	05 - 10	05 - 20
	HDU-R223	02 / 0E	05 - 04	-	05 - 14	05 - 27
	HDU-R224	02 / 0D	05 - 04	-	05 - 12	05 - 24
	HDU-R225	02 / 0B	05 - 03	05 - 04	05 - 08	05 - 17
	HDU-R226	02 / 0A	05 - 02	05 - 02	05 - 04	05 - 10
HDU-R23	HDU-R230	03 / 08	05 - 01	05 - 01	05 - 02	05 - 04
	HDU-R231	03 / 0A	05 - 02	05 - 03	05 - 06	05 - 14
	HDU-R232	03 / 0C	05 - 03	-	05 - 10	05 - 21
	HDU-R233	03 / 0E	05 - 04	-	05 - 14	05 - 28
	HDU-R234	03 / 0D	05 - 04	-	05 - 12	05 - 25
	HDU-R235	03 / 0B	05 - 03	05 - 04	05 - 08	05 - 18
	HDU-R236	03 / 09	05 - 02	05 - 02	05 - 04	05 - 11
HDU-R24	HDU-R240	04 / 09	06 - 01	06 - 01	06 - 01	06 - 05
	HDU-R241	04 / 0B	06 - 02	06 - 03	06 - 05	06 - 12
	HDU-R242	04 / 0D	06 - 03	-	06 - 09	06 - 19
	HDU-R243	04 / 08	06 - 01	-	06 - 13	06 - 01
	HDU-R244	04 / 0E	06 - 04	-	06 - 11	06 - 22
	HDU-R245	04 / 0C	06 - 03	06 - 04	06 - 07	06 - 15
	HDU-R246	04 / 0A	06 - 02	06 - 02	06 - 03	06 - 08
HDU-R25	HDU-R250	05 / 09	06 - 01	06 - 01	06 - 01	06 - 06
	HDU-R251	05 / 0B	06 - 02	06 - 03	06 - 05	06 - 13
	HDU-R252	05 / 08	06 - 01	-	06 - 09	06 - 02
	HDU-R253	05 / 0E	06 - 04	-	06 - 13	06 - 26
	HDU-R254	05 / 0D	06 - 04	-	06 - 11	06 - 23
	HDU-R255	05 / 0C	06 - 03	06 - 04	06 - 07	06 - 16
	HDU-R256	05 / 0A	06 - 02	06 - 02	06 - 03	06 - 09
HDU-R26	HDU-R260	06 / 09	06 - 01	06 - 01	06 - 02	06 - 07
	HDU-R261	06 / 08	06 - 01	06 - 03	06 - 06	06 - 03
	HDU-R262	06 / 0C	06 - 03	-	06 - 10	06 - 20
	HDU-R263	06 / 0E	06 - 04	-	06 - 14	06 - 27
	HDU-R264	06 / 0D	06 - 04	-	06 - 12	06 - 24
	HDU-R265	06 / 0B	06 - 03	06 - 04	06 - 08	06 - 17
	HDU-R266	06 / 0A	06 - 02	06 - 02	06 - 04	06 - 10
HDU-R27	HDU-R270	07 / 08	06 - 01	06 - 01	06 - 02	06 - 04
	HDU-R271	07 / 0A	06 - 02	06 - 03	06 - 06	06 - 14
	HDU-R272	07 / 0C	06 - 03	-	06 - 10	06 - 21
	HDU-R273	07 / 0E	06 - 04	-	06 - 14	06 - 28
	HDU-R274	07 / 0D	06 - 04	-	06 - 12	06 - 25
	HDU-R275	07 / 0B	06 - 03	06 - 04	06 - 08	06 - 18
	HDU-R276	07 / 09	06 - 02	06 - 02	06 - 04	06 - 11

Table 6.1 PDEV No.-C#/R# Matrixes (5/6)

HDU No	P DEV No.	C# / R#	RAID Group#			
			RAID5		RAID1	RAID0
			6D+1P	3D+1P		
HDU-L20	HDU-L200	08 / 09	07 - 01	07 - 01	07 - 01	07 - 05
	HDU-L201	08 / 0B	07 - 02	07 - 03	07 - 05	07 - 12
	HDU-L202	08 / 0D	07 - 03	-	07 - 09	07 - 19
	HDU-L203	08 / 08	07 - 01	-	07 - 13	07 - 01
	HDU-L204	08 / 0E	07 - 04	-	07 - 11	07 - 22
	HDU-L205	08 / 0C	07 - 03	07 - 04	07 - 07	07 - 15
	HDU-L206	08 / 0A	07 - 02	07 - 02	07 - 03	07 - 08
HDU-L21	HDU-L210	09 / 09	07 - 01	07 - 01	07 - 01	07 - 06
	HDU-L211	09 / 0B	07 - 02	07 - 03	07 - 05	07 - 13
	HDU-L212	09 / 08	07 - 01	-	07 - 09	07 - 02
	HDU-L213	09 / 0E	07 - 04	-	07 - 13	07 - 26
	HDU-L214	09 / 0D	07 - 04	-	07 - 11	07 - 23
	HDU-L215	09 / 0C	07 - 03	07 - 04	07 - 07	07 - 16
	HDU-L216	09 / 0A	07 - 02	07 - 02	07 - 03	07 - 09
HDU-L22	HDU-L220	0A / 09	07 - 01	07 - 01	07 - 02	07 - 07
	HDU-L221	0A / 08	07 - 01	07 - 03	07 - 06	07 - 03
	HDU-L222	0A / 0C	07 - 03	-	07 - 10	07 - 20
	HDU-L223	0A / 0E	07 - 04	-	07 - 14	07 - 27
	HDU-L224	0A / 0D	07 - 04	-	07 - 12	07 - 24
	HDU-L225	0A / 0B	07 - 03	07 - 04	07 - 08	07 - 17
	HDU-L226	0A / 0A	07 - 02	07 - 02	07 - 04	07 - 10
HDU-L23	HDU-L230	0B / 08	07 - 01	07 - 01	07 - 02	07 - 04
	HDU-L231	0B / 0A	07 - 02	07 - 03	07 - 06	07 - 14
	HDU-L232	0B / 0C	07 - 03	-	07 - 10	07 - 21
	HDU-L233	0B / 0E	07 - 04	-	07 - 14	07 - 28
	HDU-L234	0B / 0D	07 - 04	-	07 - 12	07 - 25
	HDU-L235	0B / 0B	07 - 03	07 - 04	07 - 08	07 - 18
	HDU-L236	0B / 09	07 - 02	07 - 02	07 - 04	07 - 11
HDU-L24	HDU-L240	0C / 09	08 - 01	08 - 01	08 - 01	08 - 05
	HDU-L241	0C / 0B	08 - 02	08 - 03	08 - 05	08 - 12
	HDU-L242	0C / 0D	08 - 03	-	08 - 09	08 - 19
	HDU-L243	0C / 08	08 - 01	-	08 - 13	08 - 01
	HDU-L244	0C / 0E	08 - 04	-	08 - 11	08 - 22
	HDU-L245	0C / 0C	08 - 03	-	08 - 07	08 - 15
	HDU-L246	0C / 0A	08 - 02	08 - 02	08 - 03	08 - 08
HDU-L25	HDU-L250	0D / 09	08 - 01	08 - 01	08 - 01	08 - 06
	HDU-L251	0D / 0B	08 - 02	08 - 03	08 - 05	08 - 13
	HDU-L252	0D / 08	08 - 01	-	08 - 09	08 - 02
	HDU-L253	0D / 0E	08 - 04	-	08 - 13	08 - 26
	HDU-L254	0D / 0D	08 - 04	-	08 - 11	08 - 23
	HDU-L255	0D / 0C	08 - 03	-	08 - 07	08 - 16
	HDU-L256	0D / 0A	08 - 02	08 - 02	08 - 03	08 - 09

Table 6.1 PDEV No.-C#/R# Matrixes (6/6)

HDU No	P DEV No.	C# / R#	RAID Group#			
			RAID5		RAID1	RAID0
			6D+1P	3D+1P		
HDU-L26	HDU-L260	0E / 09	08 - 01	08 - 01	08 - 02	08 - 07
	HDU-L261	0E / 08	08 - 01	08 - 03	08 - 06	08 - 03
	HDU-L262	0E / 0C	08 - 03	-	-	08 - 20
	HDU-L263	0E / 0E	08 - 04	-	-	08 - 27
	HDU-L264	0E / 0D	08 - 04	-	-	08 - 24
	HDU-L265	0E / 0B	08 - 03	-	-	08 - 17
	HDU-L266	0E / 0A	08 - 02	08 - 02	08 - 04	08 - 10
HDU-L27	HDU-L270	0F / 08	08 - 01	08 - 01	08 - 02	08 - 04
	HDU-L271	0F / 0A	08 - 02	08 - 03	08 - 06	08 - 14
	HDU-L272	0F / 0C	08 - 03	-	-	08 - 21
	HDU-L273	0F / 0E	08 - 04	-	-	08 - 28
	HDU-L274	0F / 0D	08 - 04	-	-	08 - 25
	HDU-L275	0F / 0B	08 - 03	-	-	08 - 18
	HDU-L276	0F / 09	08 - 02	08 - 02	08 - 04	08 - 11

12.6.2 Commands

This subsystem commands are classified into the following eight categories:

(1) Read commands

The read commands transfer data read from devices to channels.

(2) Write commands

The write commands transfer data from channels to devices.

(3) Search commands

The search commands follow a control command and logically search for target data.

(4) Control commands

The control commands include the seek commands that position the head at the specified cylinder and head positions. The SET SECTOR command that executes latency time processing, the LOCATE RECORD command that specifies the operation of the ECKD command, the SET FILE MASK commands that defines the permissible ranges for the write and seek operations, and the DEFINE EXTENT command that defines the permissible ranges for the write and seek operations and that defines the cache access mode.

(5) Sense commands

The sense commands transfer sense bytes and device specifications.

(6) Path control commands

The path control commands enable and disable the exclusive control of devices.

(7) TEST I/O command

The TEST I/O command transfers the specified device and its path state to a given channel in the form of DSBs.

(8) Subsystem commands

The subsystem commands include those which define cache control information in the DKCs and those which transfer cache-related information to channels.

Table 12.6.1 Command Summary (1/3)

Command Name		Command Code	
		Single Track	Multitrack
Read commands	READ INITIAL PROGRAM LOAD (RD IPL)	02	
	READ HOME ADDRESS (RD HA)	1A	9A
	READ RECORD ZERO (RD R0)	16	96
	READ COUNT,KEY,DATA (RD CKD)	1E	8E
	READ KEY,DATA (RD KD)	0E	86
	READ DATA (RD D)	06	92
	READ COUNT (RD C)	12	
	READ MULTIPLE COUNT,KEY AND DATA (RD MCKD)	5E	
	READ TRACK (RD TRK)	DE	
	READ SPECIAL HOME ADDRESS (RD SP HA)	0A	
WRITE commands	WRITE HOME ADDRESS (WR HA)	19	
	WRITE RECORD ZORO (WR R0)	15	
	WRITE COUNT,KEY,DATA (WR CKD)	1D	
	WRITE COUNT,KEY,DATA NEXT TRACK (WR CKD NT)	9D	
	ERASE (ERS)	11	
	WRITE KEY AND DATA (WR KD)	0D	
	WRITE UPDATE KEY AND DATA (WR UP KD)	8D	
	WRITE DATA (WR D)	05	
	WRITE UPDATE DATA (WR UP D)	85	
	WRITE SPECIAL HOME ADDRESS (WR SP HA)	09	
SEARCH commands	SEARCH HOME ADDRESS (SCH HA EQ)	39	B9
	SEARCH ID EQUAL (SCH ID EQ)	31	B1
	SEARCH ID HIGH (SCH ID HI)	51	D1
	SEARCH ID HIGH OR EQUAL (SCH ID HE)	71	F1
	SEARCH KEY EQUAL (SCH KEY EQ)	29	A9
	SEARCH KEY HIGH (SCH KEY HI)	49	C9
	SEARCH KEY HIGH OR EQUAL (SCH KEYD HE)	69	E9

Table 12.6.1 Command Summary(2/3)

Command Name			Command Code	
			Single Track	Multitrack
CONTROL commands	DEFINE EXTENT	(DEF EXT)	63	
	LOCATE RECORD	(LOCATE)	47	
	LOCATE RECORD EXTENDED	(LOCATE EXT)	4B	
	SEEK	(SK)	07	
	SEEK CYLINDER	(SK CYL)	0B	
	SEEK HEAD	(SK HD)	1B	
	RECALIBRATE	(RECAL)	13	
	SET SECTOR	(SET SECT)	23	
	SET FILE MASK	(SET FM)	1F	
	READ SECTOR	(RD SECT)	22	
	SPACE COUNT	(SPC)	0F	
	NO OPERATION	(NOP)	03	
	RESTORE	(REST)	17	
	DIAGNOSTIC CONTROL	(DIAG CTL)	F3	
SENSE commands	SENSE	(SNS)	04	
	READ AND RESET BUFFERED LOG	(RRBL)	A4	
	SENSE IDENTIFICATION	(SNS ID)	E4	
	READ DEVICE CHARACTERISTICS	(RD CHR)	64	
	DIAGNOSTIC SENSE/READ	(DIAG SNS/RD)	C4	
PATH CONTROL commands	DEVICE RESERVE	(RSV)	B4	
	DEVICE RELEASE	(RLS)	94	
	UNCONDITIONAL RESERVE	(UNCON RSV)	14	
	SET PATH GROUP ID	(SET PI)	AF	
	SENSE SET PATH GROUP ID	(SNS PI)	34	
	SUSPEND MULTIPATH RECONNECTION	(SUSP MPR)	5B	
	RESET ALLEGIANCE	(RST ALG)	44	
TST I/O	TEST I/O	(TIO)	00	
TIC	TRANSFER IN CHANNEL	(TIC)	X8	

Table 12.6.1 Command Summary(3/3)

Command Name			Command Code	
			Single Track	Multitrack
SUBSYSTEM commands	SET SUBSYSTEM MODE	(SET SUB MD)	87	
	PERFORM SUBSYSTEM FUNCTION	(PERF SUB FUNC)	27	
	READ SUBSYSTEM DATA	(RD SUB DATA)	3E	
	SENSE SUBSYSTEM STATUS	(SNS SUB STS)	54	
	READ MESSAGE ID	(RD MSG IDL)	4E	

Note 1: Command Reject, format 0, and message 1 are issued for commands that are not listed in this table.

Note 2: TEST I/O is a CPU instruction and cannot be specified directly. However, it appears as a command to the interface.

Note 3: TIC is a type of command but runs only on a channel. It can never be visible to the interface.

12.6.3 Power Supply Circuit Diagram

See the “Location Section” for the power supply circuit diagram.

12.6.4 Pair drive matrix for RAID1(DK308-90 or DK309-180)

Table 12.6.4 Pair drive matrix for RAID1(DK308-90 or DK309-180) (1/6)

PAIR No	PDEV No	CDEV# / RDEV#
1	HDD-R107	00 / 00
	HDD-R117	01 / 00
2	HDD-R127	02 / 00
	HDD-R137	03 / 00
3	HDD-R147	04 / 00
	HDD-R157	05 / 00
4	HDD-R167	06 / 00
	HDD-R177	07 / 00
5	HDD-R100	00 / 01
	HDD-R110	01 / 01
6	HDD-R120	02 / 01
	HDD-R130	03 / 01
7	HDD-R140	04 / 01
	HDD-R150	05 / 01
8	HDD-R160	06 / 01
	HDD-R170	07 / 01
9	HDD-R106	00 / 02
	HDD-R116	01 / 02
10	HDD-R126	02 / 02
	HDD-R136	03 / 02
11	HDD-R146	04 / 02
	HDD-R156	05 / 02
12	HDD-R166	06 / 02
	HDD-R176	07 / 02
13	HDD-R101	00 / 03
	HDD-R111	01 / 03
14	HDD-R121	02 / 03
	HDD-R131	03 / 03
15	HDD-R141	04 / 03
	HDD-R151	05 / 03
16	HDD-R161	06 / 03
	HDD-R171	07 / 03

Table 12.6.4 Pair drive matrix for RAID1(DK308-90 or DK309-180) (2/6)

PAIR No	PDEV No	CDEV# / RDEV#
17	HDD-R105	00 / 04
	HDD-R115	01 / 04
18	HDD-R125	02 / 04
	HDD-R135	03 / 04
19	HDD-R145	04 / 04
	HDD-R155	05 / 04
20	HDD-R165	06 / 04
	HDD-R175	07 / 04
21	HDD-R102	00 / 05
	HDD-R112	01 / 05
22	HDD-R122	02 / 05
	HDD-R132	03 / 05
23	HDD-L107	08 / 00
	HDD-L117	09 / 00
24	HDD-L127	0A / 00
	HDD-L137	0B / 00
25	HDD-L147	0C / 00
	HDD-L157	0D / 00
26	HDD-L167	0E / 00
	HDD-L177	0F / 00
27	HDD-L100	08 / 01
	HDD-L110	09 / 01
28	HDD-L120	0A / 01
	HDD-L130	0B / 01
29	HDD-L140	0C / 01
	HDD-L150	0D / 01
30	HDD-L160	0E / 01
	HDD-L170	0F / 01
31	HDD-L106	08 / 02
	HDD-L116	09 / 02
32	HDD-L126	0A / 02
	HDD-L136	0B / 02

Table 12.6.4 Pair drive matrix for RAID1(DK308-90 or DK309-180) (3/6)

PAIR No	PDEV No	CDEV# / RDEV#
33	HDD-L146	0C / 02
	HDD-L156	0D / 02
34	HDD-L166	0E / 02
	HDD-L176	0F / 02
35	HDD-L101	08 / 03
	HDD-L111	09 / 03
36	HDD-L121	0A / 03
	HDD-L131	0B / 03
37	HDD-L141	0C / 03
	HDD-L151	0D / 03
38	HDD-L161	0E / 03
	HDD-L171	0F / 03
39	HDD-L105	08 / 04
	HDD-L115	09 / 04
40	HDD-L125	0A / 04
	HDD-L135	0B / 04
41	HDD-L145	0C / 04
	HDD-L155	0D / 04
42	HDD-L165	0E / 04
	HDD-L175	0F / 04
43	HDD-L102	08 / 05
	HDD-L112	09 / 05
44	HDD-L122	0A / 05
	HDD-L132	0B / 05
45	HDD-R142	04 / 05
	HDD-R152	05 / 05
46	HDD-R162	06 / 05
	HDD-R172	07 / 05
47	HDD-R104	00 / 06
	HDD-R114	01 / 06
48	HDD-R124	02 / 06
	HDD-R134	03 / 06

Table 12.6.4 Pair drive matrix for RAID1(DK308-90 or DK309-180) (4/6)

PAIR No	PDEV No	CDEV# / RDEV#
49	HDD-R144	04 / 06
	HDD-R154	05 / 06
50	HDD-R164	06 / 06
	HDD-R174	07 / 06
51	HDD-L142	0C / 05
	HDD-L152	0D / 05
52	HDD-L162	0E / 05
	HDD-L172	0F / 05
53	HDD-L104	08 / 06
	HDD-L114	09 / 06
54	HDD-L124	0A / 06
	HDD-L134	0B / 06
55	HDD-L144	0C / 06
	HDD-L154	0D / 06
56	HDD-L164	0E / 06
	HDD-L174	0F / 06
57	HDD-R200	00 / 09
	HDD-R210	01 / 09
58	HDD-R220	02 / 09
	HDD-R230	03 / 08
59	HDD-R240	04 / 09
	HDD-R250	05 / 09
60	HDD-R260	06 / 09
	HDD-R270	07 / 08
61	HDD-R206	00 / 0A
	HDD-R216	01 / 0A
62	HDD-R226	02 / 0A
	HDD-R236	03 / 09
63	HDD-R246	04 / 0A
	HDD-R256	05 / 0A
64	HDD-R266	06 / 0A
	HDD-R276	07 / 09

Table 12.6.4 Pair drive matrix for RAID1(DK308-90 or DK309-180) (5/6)

PAIR No	PDEV No	CDEV# / RDEV#
65	HDD-L200	08 / 09
	HDD-L210	09 / 09
66	HDD-L220	0A / 09
	HDD-L230	0B / 08
67	HDD-L240	0C / 09
	HDD-L250	0D / 09
68	HDD-L260	0E / 09
	HDD-L270	0F / 08
69	HDD-L206	08 / 0A
	HDD-L216	09 / 0A
70	HDD-L226	0A / 0A
	HDD-L236	0B / 09
71	HDD-L246	0C / 0A
	HDD-L256	0D / 0A
72	HDD-L266	0E / 0A
	HDD-L276	0F / 09
73	HDD-R201	00 / 0B
	HDD-R211	01 / 0B
74	HDD-R221	02 / 08
	HDD-R231	03 / 0A
75	HDD-R241	04 / 0B
	HDD-R251	05 / 0B
76	HDD-R261	06 / 08
	HDD-R271	07 / 0A
77	HDD-R205	00 / 0C
	HDD-R215	01 / 0C
78	HDD-R225	02 / 0B
	HDD-R235	03 / 0B
79	HDD-R245	04 / 0C
	HDD-R255	05 / 0C
80	HDD-R265	06 / 0B
	HDD-R275	07 / 0B

Table 12.6.4 Pair drive matrix for RAID1(DK308-90 or DK309-180) (6/6)

PAIR No	PDEV No	CDEV# / RDEV#
81	HDD-L201	08 / 0B
	HDD-L211	09 / 0B
82	HDD-L221	0A / 08
	HDD-L231	0B / 0A
83	HDD-L241	0C / 0B
	HDD-L251	0D / 0B
84	HDD-L261	0E / 08
	HDD-L271	0F / 0A