ST-38-R1-121093
Sega Enterprises, LTD.

External Specification
Saturn CD Communication Interface

Doc. # ST-38-R1-121093

SEGA

(MO, 1/10/94)

ST-38-R1-121093
Sega Enterprises, LTD.

External Specification

Saturn
CD Communication Interface

Tentative 1:	8/19/93
Tentative 2:	9/10/93
Tentative 3:	11/15/93
Tentative 4:	12/7/93

(MO. 1/10/94)

Table of Contents

1 Introduction

This document is a compilation of a CD block functional overview as seen from the host side and the CD communications interface function specifications. The host program can use the CD block functions through the CD communications interface via a common software interface.

1.1 Library Configuration

Figure 1.1 shows the configuration of the CD-related libraries.

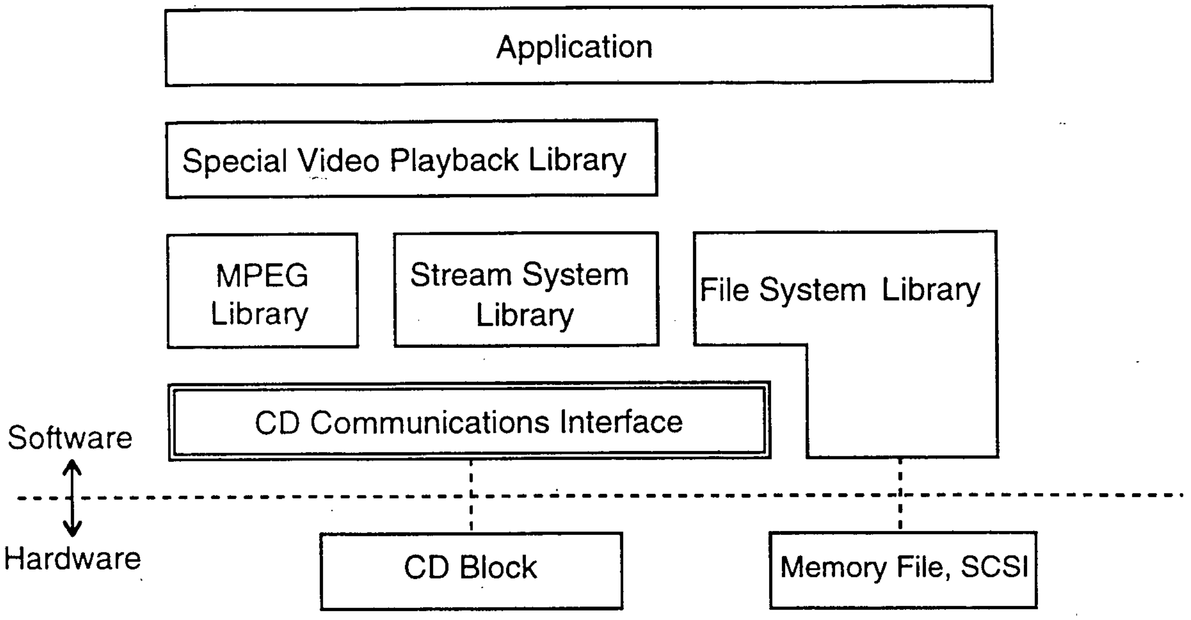


Figure 1.1 CD Library Configuration

1.2 System Configuration

Figure 1.2 shows the CD function system configuration as seen from the host side.

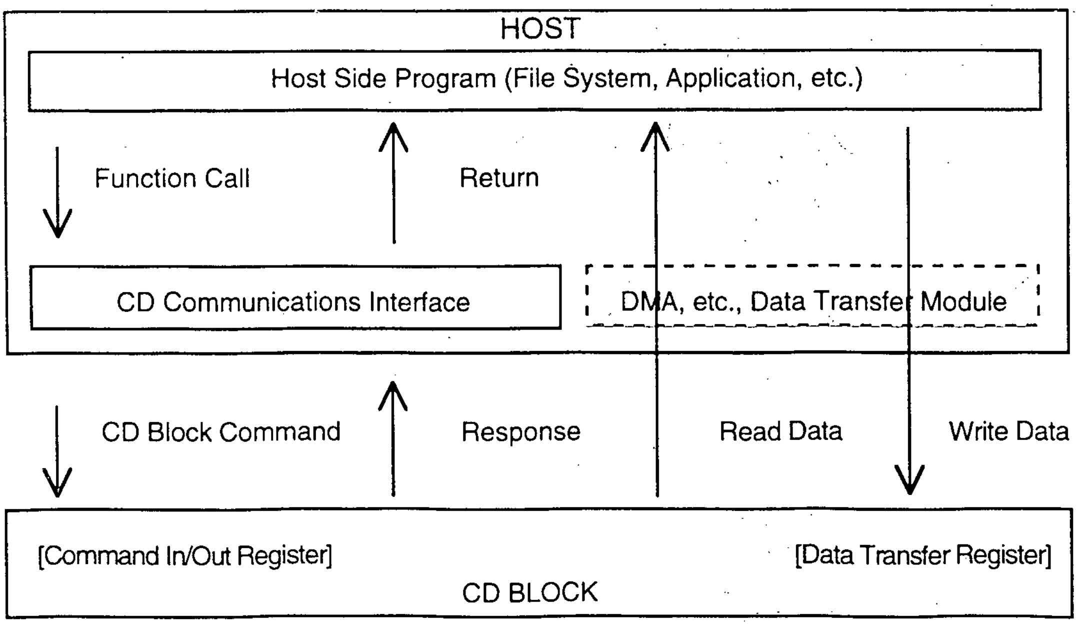

HOST
Host Side Program (File System, Application, etc.)
Function Call
Return
CD Communications Interface
DMA, etc., Data Transfer Module
CD Block Command
Response
Read Data
Write Data
[Command In/Out Register]
[Data Transfer Register]
CD BLOCK

Figure 1.2 CD Function System Configuration

1.3 CD Communications Interface Functions

(1) CD Block Communications Function

(a) C language interface function group that provides communications between the CD block and the host.
(b) Issues commands for the CD block and receives the response.
(c) Covers CD block commands mostly one for one.

(2) Support Functions for Data Transmission

Supports auxiliary functions related to data transmission such as data transmission preparation, ending, getting the transfer start address, etc.

Data that is read from or written to the CD block passes through the CD block data transfer register. The actual transfer does not use the CD communications interface. Modules such as the DMA data transmission module mediate data transfers.

(3) Register Access Function

Supports functions that access the CD block register.

2 Basic Items

2.1 Definition of Terms

Table 2.1 Definitions of Terms

Term	Definition
Sector	Basic unit of data handled by the CD block (2352 bytes fixed). Sector size and configuration follow CD-ROM format.
Stream	Classified at the sector header and subheader, this is a logically connected data flow.
CD buffer	Buffer in the CD block where sector data is stored.
CD buffer size	Size of the CD buffer sector unit.
Buffer partition	One of the logical divisions of the CD buffer into areas. Regarded as the logical element for storing and retrieving stream data
Buffer partition size	Size of the buffer partition sector unit.
Sector position	Position of the sector unit in the buffer partition. Gets a value from 0 through (buffer partition size -1). Buffer partition sector data is accessed using the sector position as a key.
Filter	Logical element that classifies the stream according to set conditions.
Selector	Made up of the aperture and buffer partition, the sector is the logical element to select a stream. You can build the selection path for the stream from the sector.
Device	Logical hardware such as CD-ROM or MPEG that creates and supplies streams. You can select streams by connecting a device to the above circuit.
Connector	Terminal to connect the aperture, buffer partition, and device.
Frame address (FAD)	Numbers are added continuously to the frame unit with the CD absolute time of 00:00:00 as 0. The CD block is accessed with the frame address as a key, not absolute time (CD-ROM sector data, CD-DA music, etc.).
Logical sector number (LSN)	With the absolute time on the CD 00:02:00 = 0, the continuous number that is attached to the sector (frame) unit. The logical sector number = frame address - 150 (2 seconds). In the directory table (ISO9660), the logical sector number is used.
File information	The directory record information saved for when the file is accessed. Included are file lead FAD, file size, unit size, gap size, file number, and attributes.

2.2 Disk Layout

(1) Track Layout

Figure 2.1 shows the relation between the track configuration and the access key that indicates the position on the disk.

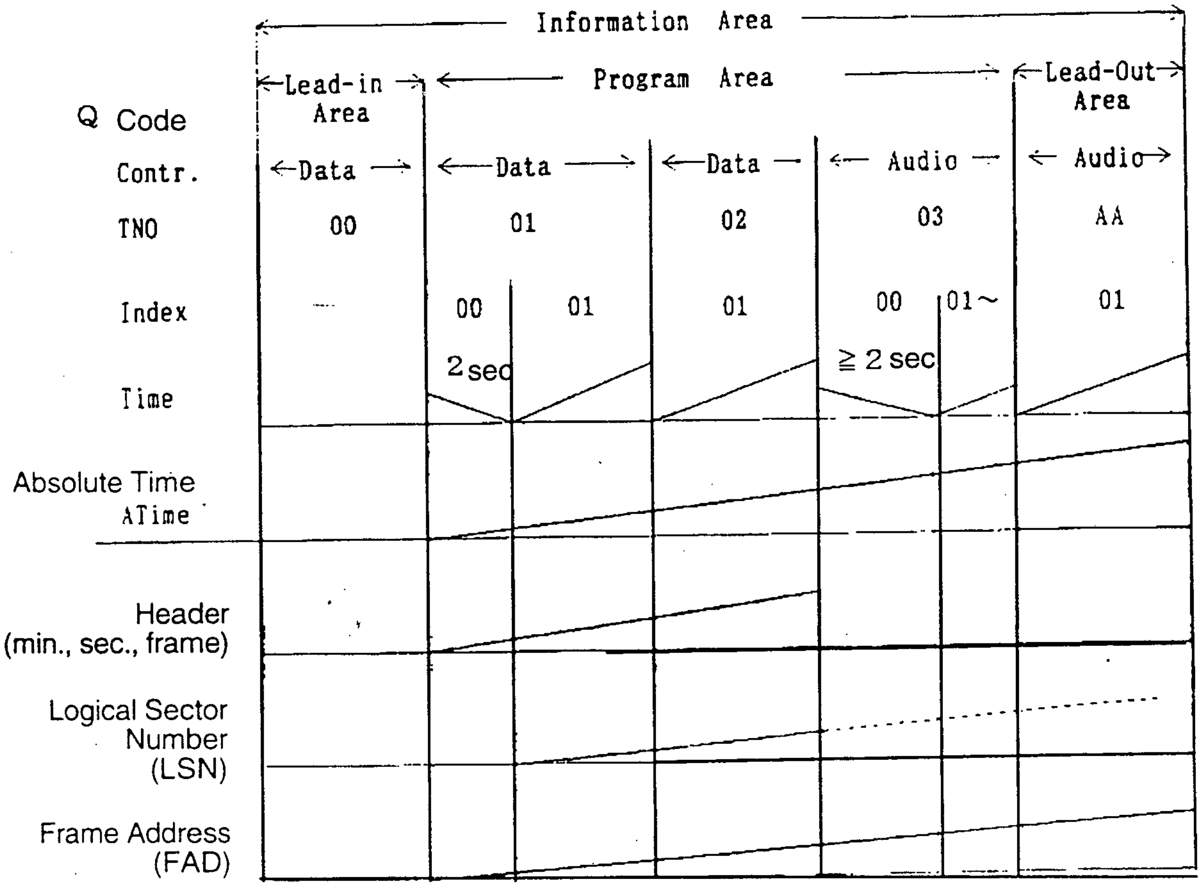


Figure 2.1 Relation of the Track Configuration and the Access Key

(2) Multisession Layout

Not documented.

3 CD Block Communications

3.1 Communications Method

(1) Start Communications Phase

Before the host can start communicating with the CD block, a check of the CD block connections is required. To do this, the Saturn IPL calls the CD block connection confirmation function (CDC_SysIsConnect). The function is only for the system, and will not be distributed for use by general users.

(2) Communications Time

The command/response time is targeted at 50 to 70µs. During this period, all interruptions are forbidden. If commands are issued continuously, the response time may suffer.

※ We are currently investigating ways to obtain the status of the CD block without using command/response.

(3) Communications Error

If the CD block does not return a response within the designated time after a command is issued, a transmission time-out error occurs. Response is detected by software loop fixed cycle polling.

3.2 Register

Table 3.1 shows a list of registers and their corresponding access functions.

Table 3.1 List of Registers

Register Name	R/W	Explanation	Access Function
DATAFIFO	R/W	Data transfer register	CDC_GetDataPtr
DATASTAT	R	Data transfer status register	CDC_GetDataStat
HIRQREQ	R/W	Interrupt cause register	CDC_GetHirqReq, CDC_ClrHirqReq
HIRQMSK	R/W	Interrupt cause mask register	CDC_GetHirqMsk, CDC_SetHirqMsk
MPEGRGB	R	MPEG register (RGB data)	CDC_GetMpegPtr

※ Access widths are all in word units (16bit).

Details of the transmission registers are shown on the next page.

In the following diagrams, unused bits are indicated by a dash (—). The initial value of unused bits is undefined. The default value of used bits is “0.”

(1) Data Transfer Register (DATATRNS)
This register inputs and outputs data between the host and the CD block during transfer.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W																

(2) Data Transfer Status Register (DATASTAT)
This register is used to confirm the status of the data transfer register.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	–	–	–	–	–	–	–	–	–	–	–	–	–	EMP	FUL	DIR

DIR (bit 0) : Data transfer direction 0=CD block to host 1=host to CD block
FUL (bit 1) : 1 = data in FIFO is full
EMP (bit 2) : 1 = data in FIFO is empty

(3) Interrupt Cause Register (HIRQREQ)
This register is the CD block-to-host interrupt cause register.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	–	–	–	–	–	–	–	–	–	MPEG	DCHG	PEND	BFUL	CSCT	DRDY	CMOK

CMOK (bit 0) : Able to issue command 1 = Response is set, can issue
0 = In a command, waiting for response
DRDY (bit 1) : Data transfer prep 1 = End preparation 0 = No preparation
CSCT (bit 2) : Interrupt 1 sector 1 = Confirm location of 1 sector read from CD-ROM
BFUL (bit 3) : Buffer full 1 = Full buffer state 0 = Buffer has space
PEND (bit 4) : End play 1 = Current FAD outside area 0 = CD is playing
DCHG (bit 5) : Change disk 1 = Disk has changed (tray is open)
MPEG (bit 6) : Prepare MPEG data 1 = End MPEG preparation 0 = Not finished
(a) Only "0" (clear) can be written to bits; "1" is not allowed.
(b) The IRQ output to the host is all of the OR output from the causes.
During interrupt processing, the appropriate bit is cleared.

(4) Interrupt cause mask register (HIRQMSK)
This register masks interrupts from the CD block to the host.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	–	–	–	–	–	–	–	–	–	MPEG	DCHG	PEND	BFUL	CSCT	DRDY	CMOK

(a) The contents of this bit are the same as those of the HIRQREQ register. (1 = Interrupt permission, 0 = Same mask).
(b) Masking the interrupt (bit = 0) causes the following when an interrupt signal is input.

- It is not reflected in the IRQ output.
- It is reflected in the HIRQREQ register (polling is possible).

(5) MPEG Register (MPEGRGB)
This register is used to transfer MPEG RGB data to the host.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																

MPEG data transfer method is not currently defined.

3.3 Data Transfer

(1) Data Transfer Procedure

The commands required for data transfer are processed by the host after the command/response. That procedure is shown below.

(a) Call the required functions for data transfer.
(b) Call the data transfer preparation function (CDC_DataReady).
(c) Data is input/output through the data transfer register (DATATRNS).
The DATATRNS register pointer (CDC_GetDataPtr) is used to get the address.
(d) The end data transfer function (CDC_DataEnd) function is called when the required data has been transferred.

(2) Data Transfer Error

After the command/response, if the CD block does not respond with data transfer preparation finished within the set time, a data transfer time-out error occurs. In this case, the host calls the data transfer end function.

4. Status Transition

4.1 Status Transition of the CD Drive

(1) Status and CD Drive Status

Responses for the different commands return the following information as status.

- REJECT: Command format is incorrect or the command itself is invalid.
- WAIT: Command was received, but has not currently been executed.
- Checks for a data transfer request.
- CD drive status.

Table 4.1 CD Drive Status

Status	Meaning
<BUSY>	Status in transition
<PAUSE>	Paused
<STANDBY>	Standby (drive is stopped)
<PLAY>	CD is playing
<SEEK>	Seeking
<SCAN>	Scanning
<OPEN>	Tray is open
<NODISC>	No disc
<RE-TRY>	Retrying read
<ERROR>	Read data error occurred
<FATAL>	A fatal error occurred (must reset hardware)

(2) Types of Drive Commands

Table 4.2 shows the types of commands used to change the status of the CD drive (drive commands).

Table 4.2 Types of Drive Commands

Command Type	Corresponding Function
Initialize the CD block (CD_INIT)	CDC_CdInit
Open Tray (OPEN_TRAY)	CDC_OpenTray
Play (CD_PLAY)	CDC_CdPlay
Seek (CD_SEEK)	CDC_CdSeek
Pause (CD_SEEK_pause)	CDC_CdSeek
Stop (CD_SEEK_home)	CDC_CdSeek
Scan (CD_Scan)	CDC_CdScan

(3) Changing the Status

The <BUSY> status is used when changing the status. The normal response to a drive command is <BUSY>.

(4) Diagram of the CD Drive Status Transition (Normal Routine)
Figure 4.1 shows the normal CD drive status transition.

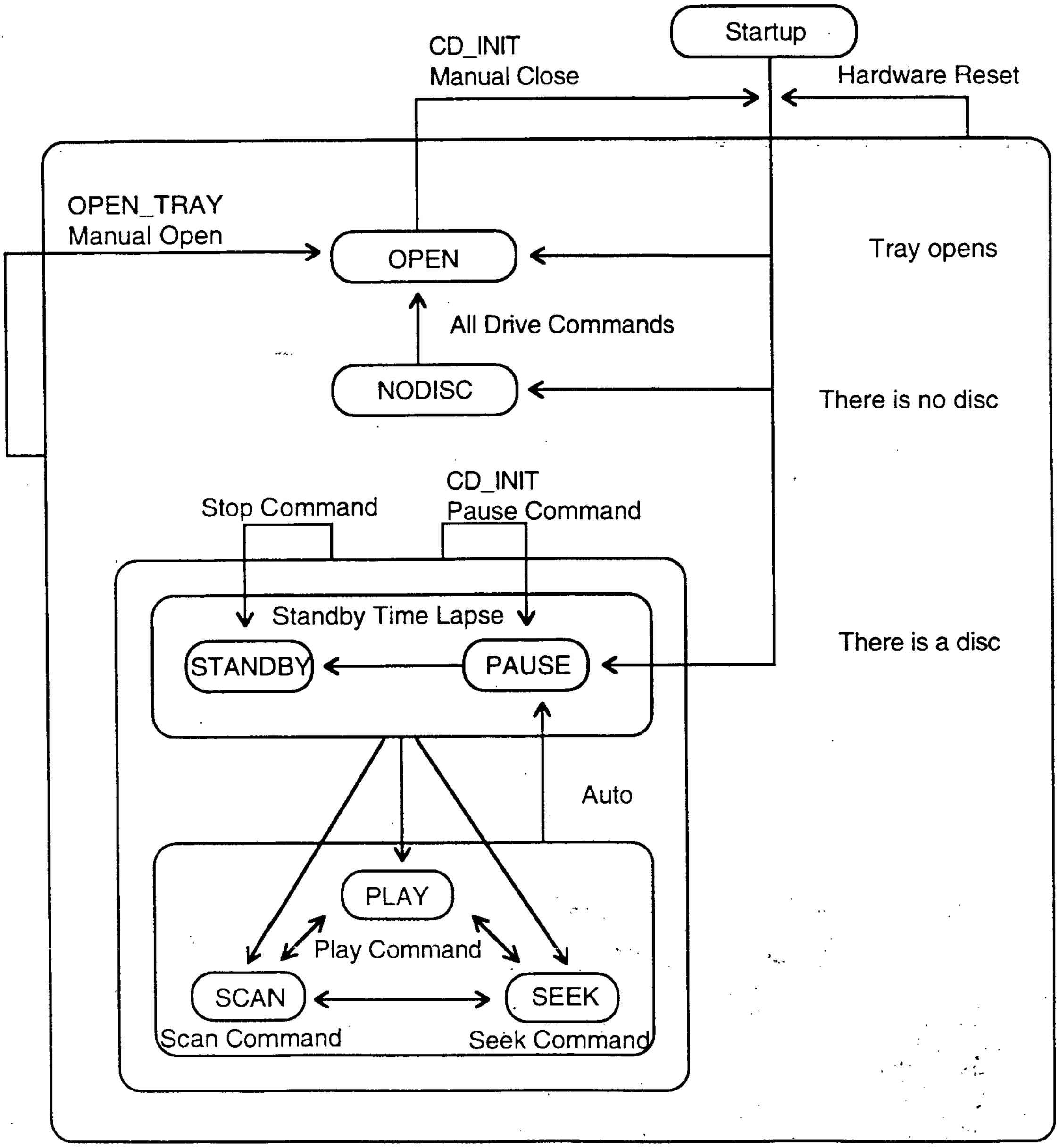


Figure 4.1 CD Drive Status Transition

(a) After startup, the drive goes into <pause> status at 2 seconds 0 frames (FAD=150) (in multisessions, 2 seconds 0 frames from the last session lead track).
(b) In transition (arrows) becomes <BUSY> status.
(c) When the tray is not an auto open/close type, it stays in the <BUSY> state until it is moved manually.
(d) Drive commands when in the <BUSY> state must WAIT.
(e) Drive commands where there are no arrows must WAIT.
(f) When making the transition to <PLAY> or <SCAN>, it may go through <SEEK>.

(5) CD Drive Status Transition Diagram (Error Routine)
Not yet documented. (To come)

4.2 CD Drive Operation

(1) Frame Address in the <PLAY> Status

The frame address during CD play (current FAD) indicates the sector in read processing. The current FAD sector is not stored in the CD buffer and therefore cannot be called up. The host can access up to the sector before the current FAD.

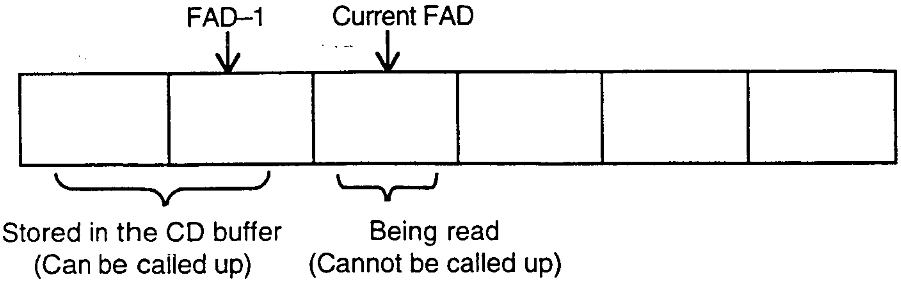


Figure 4.3 Sector Indicated by the Current FAD

If play has ended, FAD = the position where play ended + 1.
(This is valid even if the disk has ended; it merely indicates the read-out area.)

(2) Play Range and Frame Address

If the FAD play goes out of range because of a change in play end, seek, or play range, then the <PAUSE> state comes on and turns on the interrupt cause register play end flag (PEND).
If FAD is out of range and pause is disengaged (resume play), it remains in <PAUSE>.
If FAD is in range and pause is disengaged, then it starts <PLAY>.

(3) Pickup Position in the <STANDBY> State

• When moving from the <PAUSE> state:	Current position (FAD also)
• Play stopped (seeking home position):	Innermost position

(4) CD Play Repeat Processing

Both repeat notice repetitions and maximum indicated repetitions are shown in 4 bits (0 to FH).
After playing the last frame position (FAD=play end position +1), the following processing is executed.

(a) Repeats "repeat" if the number of repetitions is less than the maximum (returns to play start positions and continues play.) If repetitions are less than FH (15 times), then repetitions are increased by 1.

(b) Does not repeat if repetitions are greater than or equal to the maximum (pauses at play end position + 1).

If the play range or maximum repetitions change, the repeat repetitions are cleared to 0.
Repeat repetitions do not depend on operations such as seek during play.

(5) CD Read Operation When Buffer is Full

If the CD buffer is full, the <PAUSE> state starts and the buffer full flag of the interrupt cause register (BFUL) turns ON. When space becomes available in the buffer, the remainder is automatically played.

5 CD Block Configuration

The logical configuration and characteristics of the CD buffer as seen from the host are shown below.

(1) Data Format
The basic unit of the stream handled by the CD buffer independent of the device is a fixed sector size of 2352 bytes.

(2) Stream Selection Circuit
The functions that separate and store the streams are the aperture and buffer partition. These functions are recognized as a logical element (selector). By combining selectors, the circuit to select the required stream can be built.

(3) Device
CD, MPEG, partial RAM, etc., which create and absorb stream, are regarded as logical devices. The stream selection circuit controls the stream flow by connecting the devices.

5.1 Data Flow

Figure 5.1 shown the overall data flow related to the CD block.

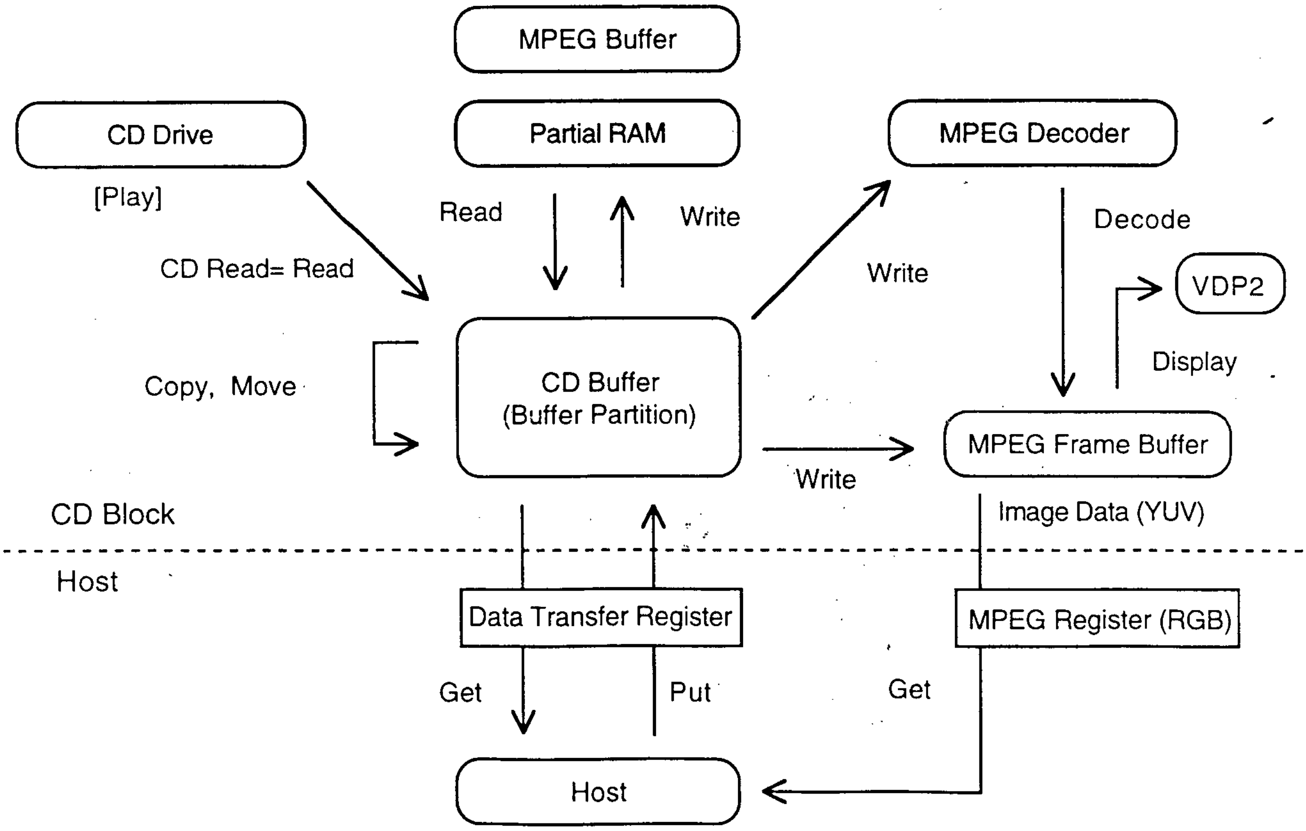


Figure 5.1 CD Block Overall Data Flow

5.2 Stream Processing Configuration

Figure 5.2 shows the CD block overall configuration as it relates to stream flow.

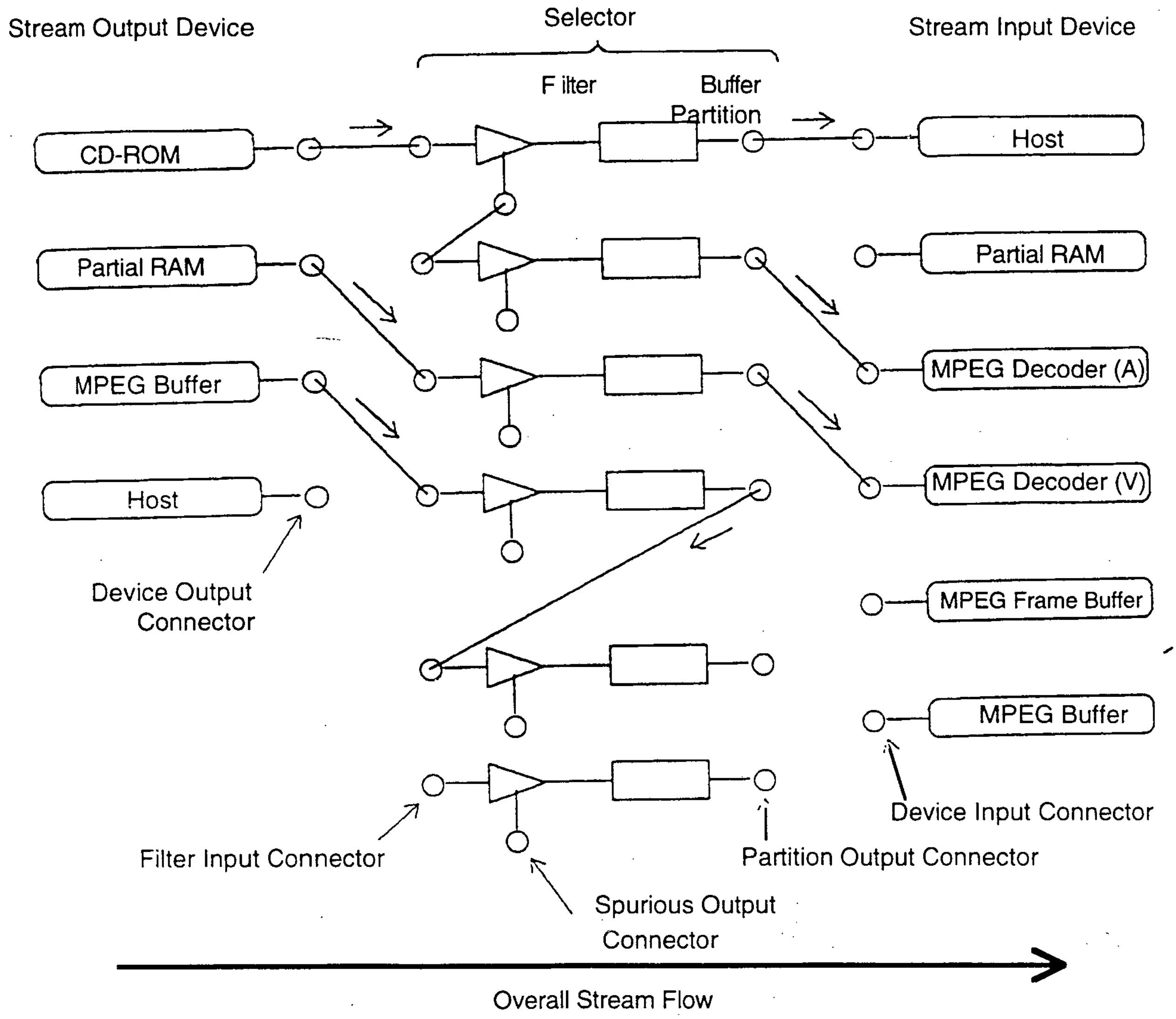


Figure 5.2 Overall Configuration of the CD Block

Explanation of Parts

(a) Stream outflow, inflow devices: Create or receive streams.
(b) Selector: Made up of the filter and buffer partition; it selects the stream. Currently there are 24 selectors, numbered 0 to 23.
(c) Filter: Separates the stream according to set conditions (matches or doesn't match).
(d) Buffer partition: Stores the stream and outputs it when there is an outside request. (As long as there is space in the total buffer, there is no capacity limit.)

Each device and selector has a connector for connections. Device and selector input and output connectors are connected one for one. Selectors can also be connected together. The stream flows regularly between connectors without passing through the buffer. If it enters the partition, it stops. The outflow and inflow streams must always be connected through a selector. The device output connector can be connected to a filter input connector, and the device input connector can be connected to the partition output connector. Selectors that are output from disconnected output connectors are deleted.

5.3 Selector Configuration

Figure 5.3 shows the selector configuration.

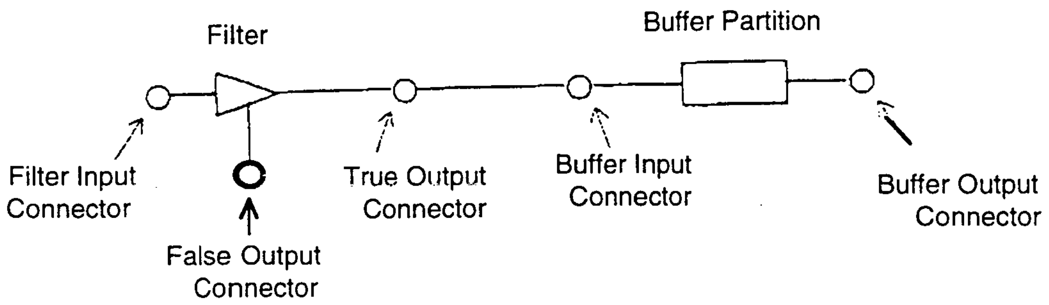


Figure 5.3 Selector Configuration

In the default selector state, the same number filter and partition are connected through the true output connector and partition output connector. Other connectors are not connected.

5.3.1 Filter

The filter is used to set the sector conditions for passing (FAD range, subheader). Sectors that meet the conditions are output through the true output connector and are stored in the connected buffer partition. Other sectors are output through the false output connector.

The false output connector can be connected to other filter input connectors and the same process repeated over and over again. Sectors that are output from unconnected output connectors are deleted.

Figure 5.4 shows the stream selection process.

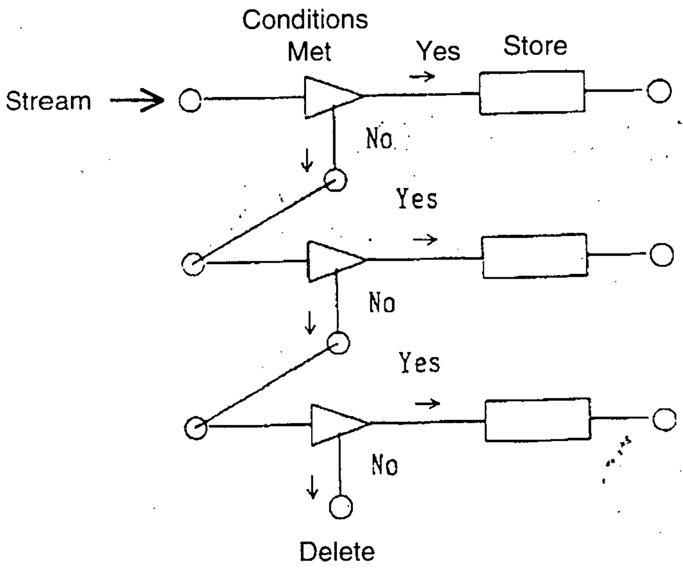


Figure 5.4 Stream Selection Process

5.3.2 Buffer Partition

(1) Storing the Sector in the Buffer Partition
The sector is stored in the last part of the partition. The last sector position in the partition depends on the size of the buffer partition. After the sector is stored, the size of the partition is increased by one only.

(2) Sector Output from an Outflow Device
Sector data is output from the outflow device one sector at a time. As soon as the selector has stored one sector in the buffer partition, the outflow device sends the next sector data. With a CD, a one-sector interrupt is created and the storage partition number is recorded. The partition number of the last sector data stored in the partition from the CD can be obtained from the host. If the sector data is thrown away, that fact is recorded.

5.3.3 Connector

(1) Sector Connection Based on OR Conditions (Multiple to One Connection)
You can connect the true output connector from multiple filters to the input connector of the same partition. This enables sectors that meet any one of several different conditions to be stored in the same sector (OR conditions).
Figure 5.5 shows selector connections according to OR conditions.

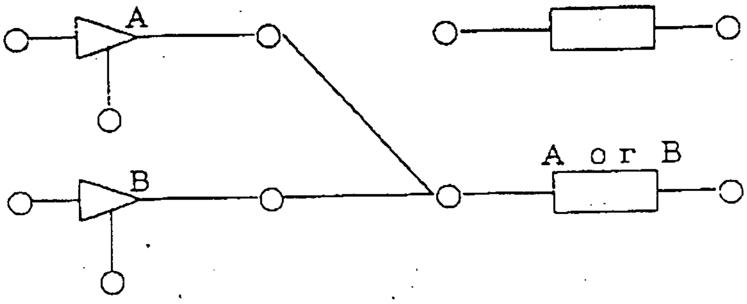


Figure 5.5 Selector Connections According to OR Conditions

(2) Connector Types and Locations
Table 5.1 shows the types of connectors and where they are connected. Only partition input can have multiple connections; others are one to one.

Table 5.1 Connection Types and Locations

Connector Type	Where Connected
Output connectors	
Device output connector	Filter input connector
True output Connector	Partition input connector (Depending on OR, can connect to same partition.)
False output connector	Filter input connector
Partition output connector	Device input connector, Filter input connector
Input connectors	
Device input connector	Partition output connector
Filter input connector	Device, false, and partition output connector
Partition input connector	True output connector (Depending on OR, multiple filters can be connected.)

5.3.4 Errors

Other than the device, stream processing errors occur when the connection is cut during operation or when the CD buffer becomes full.

(1) If, the set/get command is issued, after the validation timing for the selector is set, WAIT is returned.

(2) Sector Data During Connection Changeover
Data is not lost even if the connection is cut while the CD is reading and sector data is flowing. In other words, data is saved during changeover or temporary disconnections. If the connection is completely cut off, the stream stops.

5.4 Sector Data Format

(1) Basic Format
The format is based on the CD-ROM XA format.

(2) Handling Subheaders and User Data (2048 Bytes)
(a) Subheaders are 0 in other than mode 2 (form 1, 2).
(During initialization, if subheaders are not recognized, they are still 0.)
(b) Only in mode 1 is user data directly available to the header.
Other than that, user data is regarded as being in the same position as mode 2 form 1.
(c) If user data from a device other than a CD is stored in the buffer partition, it is treated as mode 2 form 1. The lead 24 bytes are 0. The end of user data is undefined.

5.5 Initialization of the CD Block

(1) TOC/ Session Information
Initialized when the tray is opened; no input information.
When the tray is closed, starts TOC read. (Soft reset is ignored.)

(2) File Information
Initialized when the tray is opened or a soft reset occurs; no input information.
Reading file information is clearly commanded by the host.

(3) Host Information
Initialized by a soft reset. (Tray opening is ignored.)

Host Information: Mainly host settings and buffer data.

- Initialization information (parameters of the CD block initialization function)
- Play information (parameters of the CD play function: play range, play mode)
- Selector information (host settings such as filter, buffers, etc., related to the selector.)
- Data in the buffer, buffer partition size, and empty size
- MPEG information
- Data transfer control register (data transfer ends when initialized.)

Even if the CD block initialization function is called, information in (1) through (3) is not initialized if the soft reset is not used.

6. File System of the CD Block

6.1 File Control in the CD-ROM

Figure 6.1 shows the data configuration of file control in CD-ROM (ISO9660).

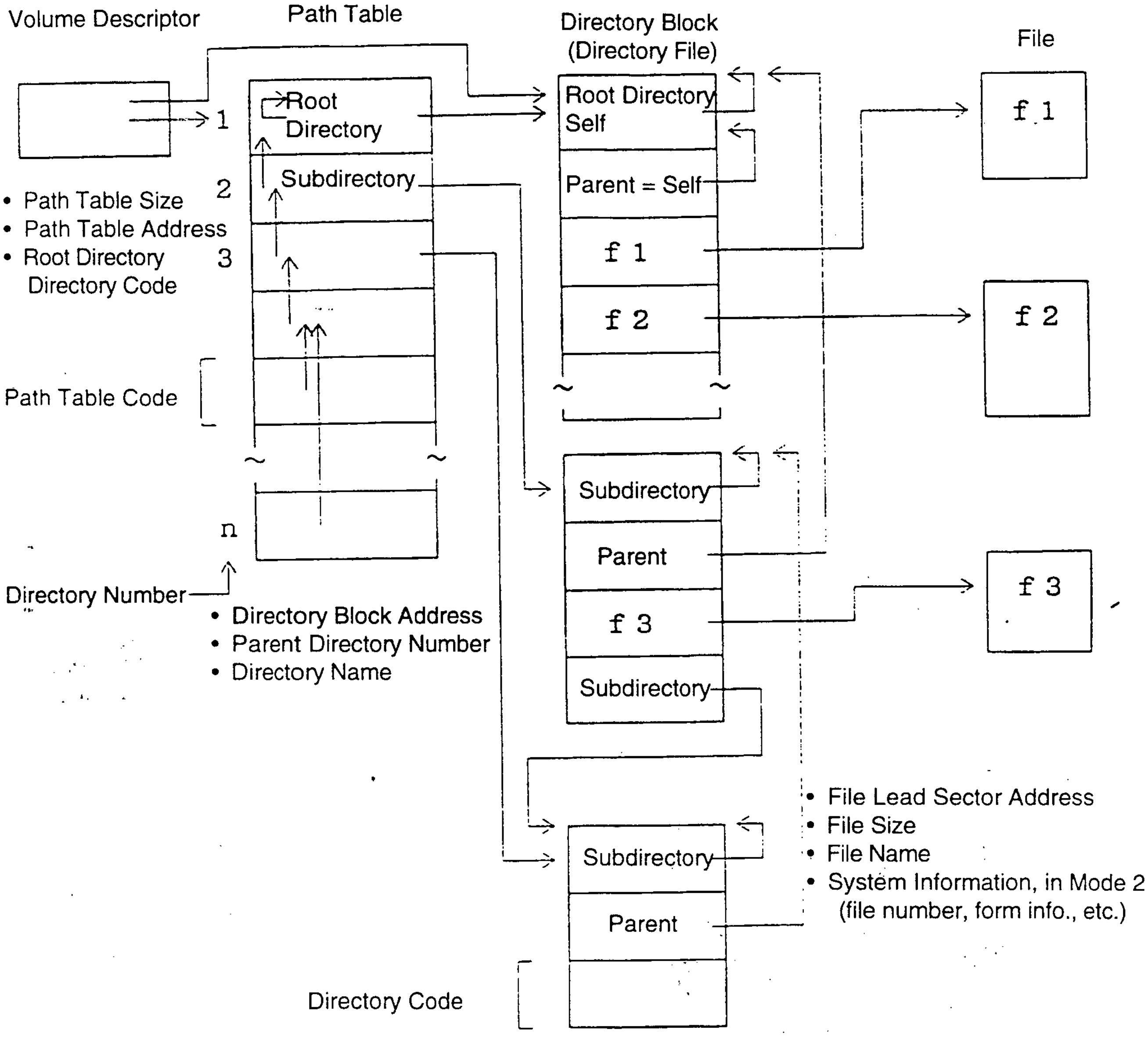


Figure 6.1 Data Configuration Of File Control In CD-ROM

• Volume descriptor:	Placed from CD 00:02:16 (FAD=166).
• Path table:	Grouping of the path table code. Path table size and address are recorded in the volume descriptor.
• Path table code:	Corresponds to one directory (root, sub). The record position becomes the directory number (1 to n). Directory information is recorded in the directory block.
• Directory block:	Grouping of the directory records. Regards the directory block as one file (directory file).
• Directory record:	Corresponds to one file (including the directory).

6.2 CD Block File System Function

Not yet documented.

7 Data Specifications

7.1 Types of Data

Table 7.1 gives CD communications interface data.

Table 7.1 Data List

Data		Data Name	No.
Basic Data			1.0
Constant			2.0
Data Type			3.0
	CD communication return code	CdcRet	3.1
	CD status information	CDcStat	3.2
	TOC information, session information	CdcToc, CdcSes	3.3
	CD position parameter	CdcPos	3.4
	CD play parameter	CdcPly	3.5
	Subheader conditions	CdcSubh	3.6
	Sector information	CdcSct	3.7
	File information	CdcFile	3.8

7.2 Data Details

Data details describe the constants, data types, access macros, etc., that are used by the CD communications interface.

7.2.1 Basic Data

Title	Data	Data Name	No
Data Specification	Basic Data		1.0

(1) Basic Data Types

Type	Explanation
Uint8	1-byte integer with no flag
Sint8	1-byte integer with flag
Uint16	2-byte integer with no flag
Sint16	2-byte integer with flag
Uint32	4-byte integer with no flag
Sint32	4-byte integer with flag
Bool	Logical type has the following values: FALSE, OFF: false TRUE, ON: true

(2) Basic Constants

Constant Name		Value	Explanation
Logical Constant		0, 1	Used as Bool type values
	FALSE	0	Shows the logical false
	TRUE	1	Shows the logical true
	OFF	0	Shows switch off (false)
	ON	1	Shows switch on (true)

7.2.2 Constants

Title	Data	Data Name	No
Data Specification	Constants		2.0

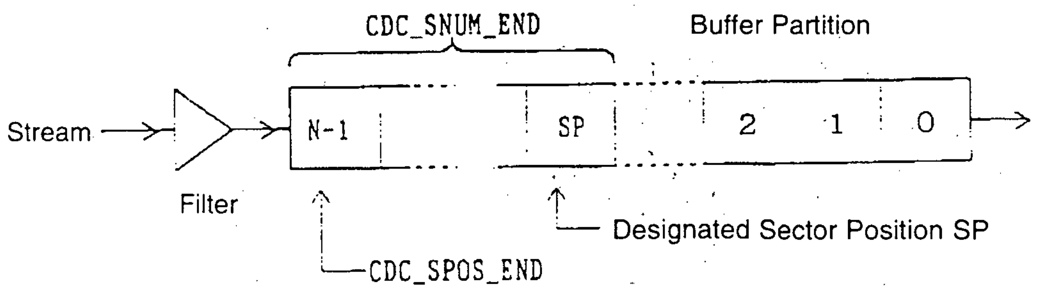
(1) Error Codes

Constant Name	Explanation
CDC_ERR_OK	Normal (process intermission).
CDC_ERR_FIN	End processing.
CDC_ERR_REJECT	The command was rejected.
CDC_ERR_WAIT	Became WAIT when the command was issued.
CDC_ERR_COMNG	CMOK flag is not 1.
CDC_ERR_TMOUT	Time-out (during response or data transfer preparation).
CDC_ERR_POSTYPE	Play or seek position type is not normal.

(2) Special Designations of the Buffer Partition Sector Range

Constant Name	Explanation
CDC_SPOS_END	Indicates the last sector in the partition. Same as the designation partition sectors -1.
CDC_SNUM_END	Shows sectors above the designated sector position SP (including SP). Same as the designation partition sectors -SP.

Valid with the following buffer partition functions.
CDC_GetActSiz CDC_GetSctInfo CDC_GetSctData CDC_DelSctData
CDC_GetdelSctData CDC_CopySctData CDC_MoveSctData



(3) Other

Constant Name	Explanation
CDC_NUL_SEL	Special value of the filter and buffer partition numbers
CDC_NUL_FID	Special value of the file identifier

7.2.3 Data Types

Title	Data	Data Name	No
Data Specification	Return Codes of the CD Communications Functions	CdcRet	3.1

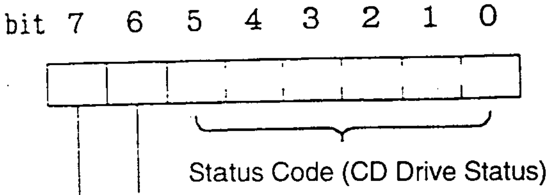
(1) Return Code

CdcRet ret ※ Careful, ret is not a pointer.

Access Macro	Type	Explanation
CDC_RET_ERR(ret)	Uint16	Error code
CDC_RET_STATUS(ret)	Uint8	Status information

(2) Status

When all bits are 1 (0xff), it indicates REJECT.



Value	Status	Meaning
0x00	<BUSY>	Status in transition
0x10	<PAUSE>	Paused
0x11	<STANDBY>	Standby (drive is stopped)
0x20	<PLAY>	CD is playing
0x21	<SEEK>	Seeking
0x22	<SCAN>	Scanning
0x30	<OPEN>	Tray is open
0x31	<NODISC>	No disc
0x32	<RETRY>	Retrying read
0x33	<ERROR>	Read data error occured
0x34	<FATAL>	A fatal error occured (must reset hardware)

1: Data transfer requested 0: Data transfer not requested

1:WAIT 0:ACK (Command acknowledged)

Title	Data	Data Name	No
Data Specification	CD Status Information	CdcStat	3.2

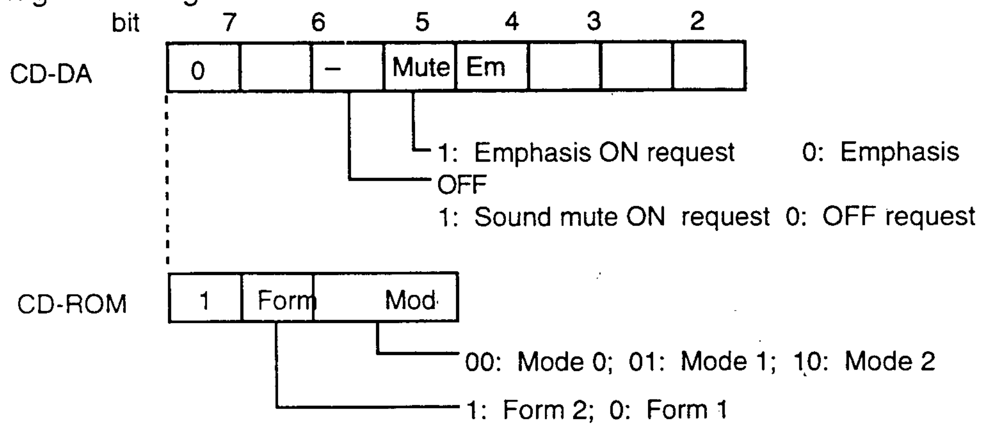
The CD block returns the CD status information as a response to the process request.

(1) CD Status Information

CdcStat *stat

Access Macro	Type	Explanation
CDC_STAT_STATUS(stat)	Uint8	Status; same as return code.
CDC_STAT_CDFLAG(stat)	Uint8	CD flag.
CDC_STAT_REPEAT(stat)	Uint8	Repeat count; recognized range is 0 to 15.
CDC_STAT_CTLADR(stat)	Uint8	CONTROL/ADR byte of subcode Q.
CDC_STAT_TNO(stat)	Uint8	2-decimal value of TNO of subcode Q.
CDC_STAT_IDX(stat)	Uint8	2-decimal value of X of subcode Q.
CDC_STAT_FAD(stat)	Sint32	Frame address.

CD Flag: 4-bit flag. Different for CD-DA and CD-ROM.



Title Data Specification	Data TOC Information, Session Information	Data Name CdcToc, CdcSes	No 3.3

(1) TOC Information (4 bytes)

CdcToc *toc

Access Macro	Type	Explanation
CDC_TOC_BDAT(toc,i)	Uint8	Byte data of the i byte offset of the TOC information (i=0 to 3)

(2) Session Information (4 bytes)

CdcSes *ses

Access Macro	Type	Explanation
CDC_SES_BDAT(ses,i)	Uint8	Byte data of the i byte offset of the session information (i=0 to 3)

Title Data Specification	Data CD Position Parameters	Data Name CdcPos	No 3.4

Designates the play range for CD play or the seek position.

(1) Position Type

The position parameter is designated by one of two methods, the frame address or the track/index. Select the method by designating the constants shown below.

Constant Name	Explanation
CDC_PTYPE_DEF	Skipped value during CD play shows disk lead/end. Skipped value during seek shows home position.
CDC_PTYPE_FAD	Designates the frame address.
CDC_PTYPE_TNO	Designates track/index.
CDC_PTYPE_NOCHG	No change to the set value.

(2) Position Parameters

CdcPos *pos

Access Macro	Type	Explanation
CDC_POS_PTYPE(pos)		Position type
CDC_POS_FAD(pos)	Sint32	Frame address, sector count
CDC_POS_TNO(pos)	Uint8	Track number
CDC_POS_IDX(pos)	Uint8	Index number

(3) Setting

(a) Designating the Frame Address

```
CdcPos pos:
    CDC_POS_PTYPE(&pos) = CDC_PTYPE_FAD;  /*designates frame address*/
    CDC_POS_FAD(&pos)   = fad;            /*start frame address*/
```

※ To designate the play end position, use the number of sectors from the start frame address.

(b) Designating the Track/Index

```
CdcPos pos:
    CDC_POS_PTYPE(&pos) = CDC_PTYPE_TNO;  /*designates track/index*/
    CDC_POS_TNO(&pos)   = tno;            /*start track number*/
    CDC_POS_IDX(&pos)   = idx;            /*start index number*/
```

※ When index = 0, only the track is indicated.

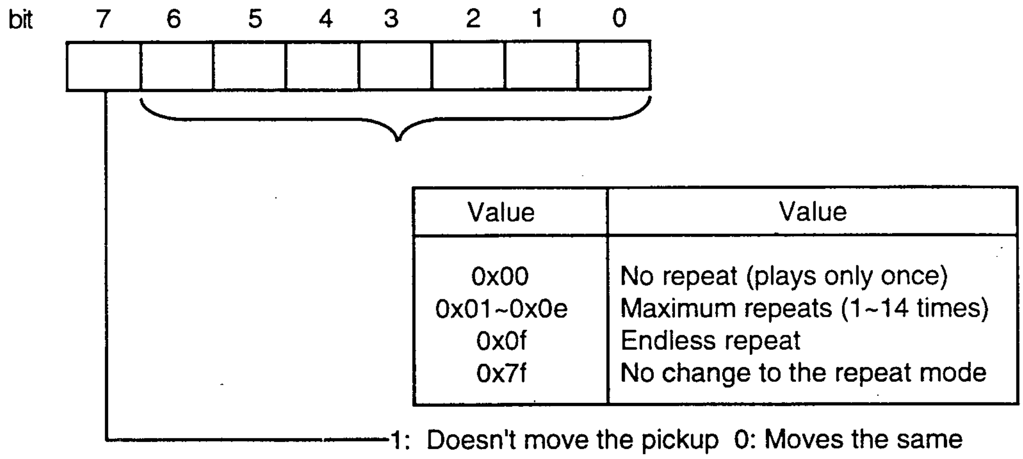
- During CD play: Shows track lead/end (same as index = 1/99).
- During seek: Shows track lead (Same as index = 1).

Title	Data	Data Name	No
Data Specification	CD Play Parameters (1)	CdcPly	3.5

Used during CD play.

(1) Play Mode

Designates movement of the pickup (play position) and the maximum repetitions for a play area during CD play.
The designation range for maximum repeats is 4 bits.



Value	Value
0x00	No repeat (plays only once)
0x01~0x0e	Maximum repeats (1~14 times)
0x0f	Endless repeat
0x7f	No change to the repeat mode

1: Doesn't move the pickup 0: Moves the same

Constant Name	Explanation
CDC_REPEAT_NOCHG	Doesn't change the maximum repeats.
CDC_PICK_MOV	Moves the pickup.
CDC_PICK_NOMOV	Doesn't move the pickup.
CDC_PMODE_NOCHG	Doesn't change the play mode. Similar to (CDC_REPEAT_NOCHG I CDC_PICK_NOMOV).

(2) Play Parameter

CdcPly *ply

Access Macro	Type	Explanation
CDC_PLY_START(ply)	CdcPos	Play start position parameters
CDC_PLY_END(ply)	CdcPos	Play end position parameters
CDC_PLY_PMODE(ply)	Uint8	Play mode (pickup movement, maximum repetitions)
CDC_PLY_STYPE(ply)		Start position type
CDC_PLY_SFAD(ply)	Sint32	Start position frame address
CDC_PLY_STNO(ply)	Uint8	Start position track number
CDC_PLY_SIDX(ply)	Uint8	Start position index number
CDC_PLY_ETYPE(ply)		End position type
CDC_PLY_EFAD(ply)	Sint32	End position frame address
CDC_PLY_ETNO(ply)	Uint8	End position track number
CDC_PLY_EIDX(ply)	Uint8	End position index number

Title	Data	Data Name	No
Data Specification	CD Play Parameters (2)	CdcPly	3.5

(3) Setting Play Parameters

In the play parameters you set the play range and play mode. The play range is designated by a combination of the position parameters.

(a) In the play range you cannot combine track/index designation and frame address/sector designations. Other than that, all combinations are valid.
(b) The play range is recorded and stored in the CD block and remains valid until it is reset.
(c) The default is from the start of the disc to the end.
(d) If you set the play mode to CDC_PMODE_NOCHG, only the play range is changed.

The following table gives some examples.

No	Play Method	Start Position	End Position	Comments
1	Track number designation.	tno1, x1	tno2, x2	
2	Frame address designation.	fad	fasnum	*1
3	Play from disc start.	CDC_PTYPE_DEF	tno2, x2 fasnum	*2
4	Play to disc end.	tno1, x1 fad	CDC_PTYPE_DEF	
5	Change only the start position.	tno1, x1 fad	CDC_PTYPE_NOCHG	
6	Change only the end position.	CDC_PTYPE_NOCHG	tno2, x2 fasnum	*3
7	Start play from the current position without changing the play range or the play mode.	CDC_PTYPE_NOCHG	CDC_PTYPE_NOCHG	Designates CDC_PMODE_NOCHG as the play mode.

fad: frame address, fasnum:	frame address sector count
tno1: start track number, x1:	start index number
tno2: start track number, x2:	start index number

{Translator's note: Regarding the above values for tno1 and tno2: although tno2 is referred to as the end position of track number 1 in the above table, here it is referenced as the start track number for track number 2. We believe both definitions to be correct.}

*1: fad to fad+fasnum -1 are stored as the play range.
*2: Disc start to disc start+fasnum -1 are stored as the play range.
*3: If the start position is not changed, the pickup does not move despite the play mode setting.

Title Data Specification	Data Sub Header Conditions	Data Name CdcSubh	No 3.6

(1) Subheader Conditions

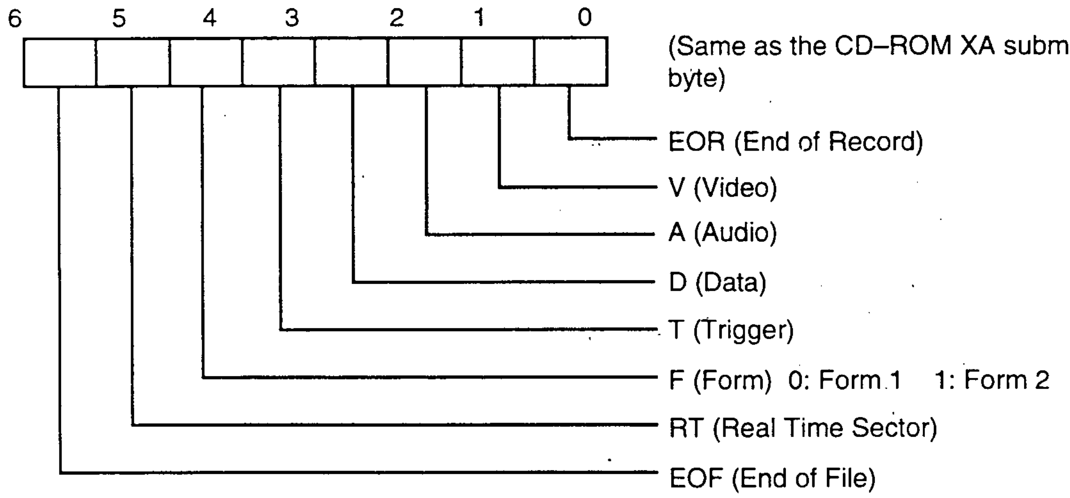
CdcSubh *subh

Access Macro	Type	Explanation
CDC_SUBH_FN(subh)	Uint8	File number
CDC_SUBH_CN(subh)	Uint8	Channel number
CDC_SUBH_SMMSK(subh)	Uint8	Submode mask pattern (comparative object bit pattern)
CDC_SUBH_SMVAL(subh)	Uint8	Submode comparator (dot comparator designated by the mask pattern.)
CDC_SUBH_CIMSK(subh)	Uint8	Coding information mask pattern (comparative object bit pattern)
CDC_SUBH_CIVAL(subh)	Uint8	Coding information mask pattern (dot comparator designated by the mask pattern.)

After the logical results of the submode and coding information are taken, these are determined to be comparative or related.

Condition Equation
(sm & SMMSK) == SMVAL
(ci & CIMSK) == CIVAL

(2) Submode



(Same as the CD–ROM XA submode byte)

Constant Name	Value	Explanation
CDC_SM_EOR	0x01	Record end sector
CDC_SM_VIDEO	0x02	Video sector
CDC_SM_AUDIO	0x04	Audio sector
CDC_SM_DATA	0x08	Data sector
CDC_SM_TRIG	0x10	Trigger bit
CDC_SM_FORM	0x20	Form bit
CDC_SM_RT	0x40	Real time sector
CDC_SM_EOF	0x80	File end sector

Title Data Specification	Data Sector Information	Data Name CdcSct	No 3.7

(1) Sector Information

CdcSct *sct

Access Macro	Type	Explanation
CDC_SCT_FAD(sct)	Sint32	Frame address
CDC_SCT_FN(sct)	Uint8	File number
CDC_SCT_CN(sct)	Uint8	Channel number
CDC_SCT_SM(sct)	Uint8	Submode
CDC_SCT_CI(sct)	Uint8	Coding information

Title	Data	Data Name	No
Data Specification	File Information	CdcFile	3.8

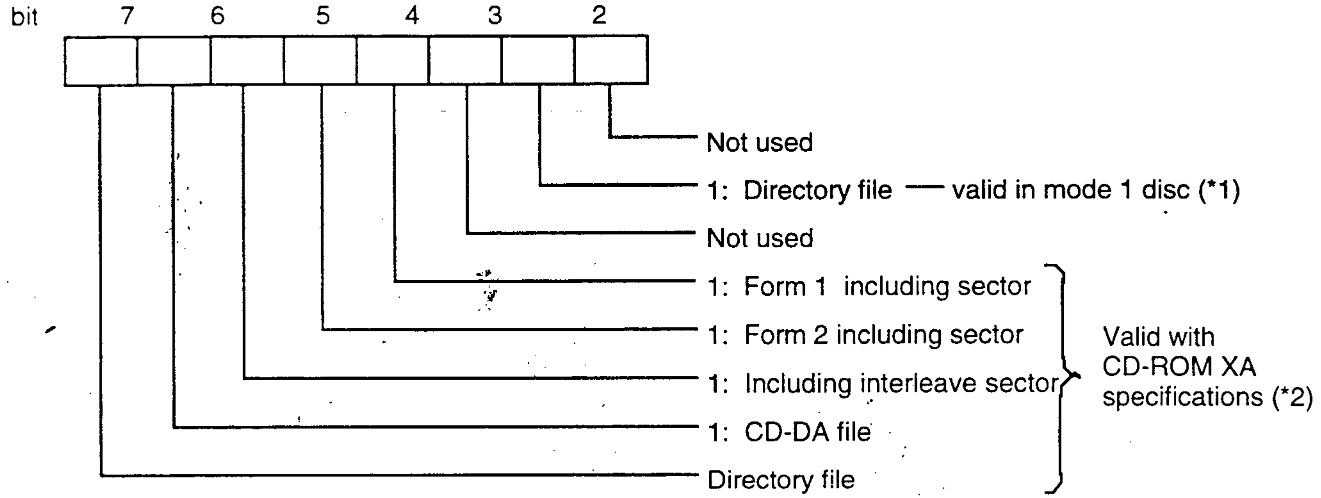
(1) File Information

CdcFile *file

Access Macro	Type	Explanation
CDC_FILE_FAD(file)	Sint32	File start frame address
CDC_FILE_SIZE(file)	Sint32	File size (bytes)
CDC_FILE_UNIT(file)	Uint8	File unit size
CDC_FILE_GAP(file)	Uint8	Gap size
CDC_FILE_FN(file)	Uint8	File number (*1)
CDC_FILE_ATR(file)	Uint8	File attributes

*1 The file number is 0 for a mode 1 disc.

(2) File Attributes


bit
7
6
5
4
3
2
Not used
1: Directory file — valid in mode 1 disc (*1)
Not used
1: Form 1 including sector
1: Form 2 including sector
1: Including interleave sector
1: CD-DA file
Directory file
Valid with CD-ROM XA specifications (*2)

*1: Bit1 is the directory bit of the file flag in the directory record and is valid in a mode 1 disc.
*2: Bit3 to 7 are attribute information based on CD_ROM XA specification; with a mode 1 disc, bits 3 to 7 are all 0.

8 Function Specifications

(1) Items of Special Mention

(a) All CD play functions are standardized without separating music play (CD-DA) and data read (CD_ROM). This is different from the normal CD system, so be careful.

(b) As a rule, BCD is not used. Track numbers, etc., that are recorded in BCD are handled as binary values.

(2) Cautions on Expressions

Status and report refresh functions are indicated as shown below.

```
[SR]  S: Refresh status
[S-]  R: Refresh report
[--]  -: Do not refresh
```

Example of a Function Title

Title	Function	Function name [SR]	No
Function specification	CD play	CDC_CdPlay	2.1

8.1 List of Functions

Table 8.1 shows a list of functions for the CD communications interface.

Table 8.1 CD Communications Interface Functions (1)

Function	Function Name	No.
Common CD Block		1.0
Get current CD status	CDC_GetCurStat	1.1
Get last CD status	CDC_GetLastStat	1.2
Get hardware information	CDC_GetHwInfo	1.3
Get TOC information	CDC_GetToc	1.4
Get session information	CDC_GetSes	1.5
Initialize the CD block	CDC_CdInit	1.6
Open tray	CDC_OpenTray	1.7
Data ready to transfer	CDC_DataReady	1.8
End data transfer	CDC_DataEnd	1.9
CD Drive		2.0
Play CD	CDC_CdPlay	2.1
Seek play position	CDC_CdSeek	2.2
Fast forward play	CDC_CdScan	2.3
Subcode		3.0
Get subcode Q	CDC_GetScdQch	3.1
Get subcode R to W	CDC_GetScdRwch	3.2
CD-ROM Device		4.0
Connect CD device	CDC_CdSetCon	4.1
Get connect status of the CD device	CDC_CdGetCon	4.2
Get last store sector in the buffer partition	CDC_CdGetLastBuf	4.3
Selector		5.0
Set filter sector range	CDC_SetFiltRange	5.1
Get filter sector range	CDC_GetFiltRange	5.2
Set filter subheader status	CDC_SetFiltSubh	5.3
Get filter subheader status	CDC_GetFiltSubh	5.4
Set filter mode	CDC_SetFiltMode	5.5
Get filter mode	CDC_GetFiltMode	5.6
Set filter connection	CDC_SetFiltCon	5.7
Get filter connection	CDC_GetFiltCon	5.8
Reset selector (filter partition)	CDC_ResetSelector	5.9

(Continues)

Table 8.1 CD Communications Interface Functions (Continued)

Function	Function Name	No.
Buffer Information		6.0
Get CD buffer size	CDC_GetBufSiz	6.1
Get buffer partition data count	CDC_GetSctNum	6.2
Get actual data size	CDC_GetActSiz	6.3
Get sector information	CDC_GetSctInfo	6.4
Buffer Input / Output		7.0
Set sector length	CDC_SetSctLen	7.1
Get sector data	CDC_GetSctData	7.2
Delete sector data	CDC_DelSctData	7.3
Get and delete sector data	CDC_GetdelSctData	7.4
Write sector data	CDC_PutSctData	7.5
Copy sector data	CDC_CopySctData	7.6
Move sector data	CDC_MoveSctData	7.7
Get sector data copy/move error	CDC_GetPlacErr	7.8
CD Block File System		8.0
Move directories	CDC_ChgDir	8.1
Directory retains file information	CDC_ReadDir	8.2
Get file information range	CDC_GetFileScope	8.3
Get file information	CDC_GetFileInfo	8.4
Get special file information	CDC_GetOneFileInfo	8.5
Start file read	CDC_ReadFile	8.6
Access Register		9.0
Get data transfer register pointer	CDC_GetDataPtr	9.1
Get value of the file transfer status register	CDC_GetDataStat	9.2
Get value of the interrupt cause register	CDC_GetHirqReq	9.3
Clear the interrupt cause register	CDC_ClrHirqReq	9.4
Get value of the interrupt cause mask register	CDC_GetHirqMsk	9.5
Set value of the interrupt cause mask register	CDC_SetHirqMsk	9.6
Get the MPEG register pointer	CDC_GetMpegPtr	9.7
System Functions (system functions not released)		10.0
Confirm CD block connection	CDC_SysIsConnect	10.1
Direct issue of CD block command	CDC_SysPrimCmd	10.2

8.2 Function Details

8.2.1 Common CD Block

Title Function Specification	Function Get Current CD Status	Function Name [SR] CDC_GetCurStat	No 1.1

[Format] CdcRet CDC_GetCurStat(CdcStat *cdstat)
[Input] None
[Output] cdstat : CD Status
[Function Value] Returns a return code.
[Function]
Issues a get status command for the CD block and gets the current status or report.

Title Function Specification	Function Get Last CD Status	Function Name [--] CDC_GetLastStat	No 1.2

[Format] CdcRet CDC_GetLastStat(CdcStat *cdstat)
[Input] None
[Output] cdstat : CD Status
[Function Value] Returns a return code.
[Function]
Gets the status and report for the last CD block command.
The CD communications interface issues a CD block command; the response, status, and report are stored. This command gets those stored values.

Title Function Specification	Function Get Hardware Information	Function Name [S-] CDC_GetHwInfo	No 1.3

[Format] CdcRet CDC_GetHwInfo(Sint32 *ver, Sint32 *hwflg, Sint32 *prsiz
[Input] None
[Output] ver : CD block version (0x00 to 0xff)
hwflg : Hardware flag (bit 0: connected to a CD emulator)
prsiz : Size of the partial RAM (2048-byte sector unit)
[Function Value] Returns a return code. .
[Function]
Gets the hardware information of the CD block.

Title Function Specification	Function Get TOC Information	Function Name [S-] CDC_GetToc	No 1.4

[Format] CdcRet CDC_GetToc(CdcToc *toc)
[Input] None
[Output] toc : TOC information stored range (408 bytes)
[Function Value] Returns a return code.
[Function]
Gets all TOC information (102 items).
The stored range must be secured as follows:

```
    CdcToc  toc[102];
```

Title Function Specification	Function Get Session Information	Function Name [S-] CDC_GetSes	No 1.5

[Format] CdcRet CDC_GetSes(Sint32 sesno, CdcSes *ses)
[Input] sesno : Session number (0 to 99)
[Output] ses : Session information (4 bytes)
[Function Value] Returns a return code.
[Function]
Gets the session information from the indicated session (multisession CD information).

Title	Function	Function Name [SR]	No
Function Specification	Initialize CD Block	CDC_CdInit	1.6

[Format] CdcRet CDC_CdInit(Sint32 initflg, Sint32 stbytim, Sint32 eccnum,
Sint32 rtrynum)

[Input] initflg : Initialize flag
stbytim : Standby time
eccnum : ECC repetitions
rtynum : Retry repetitions

[Output] None

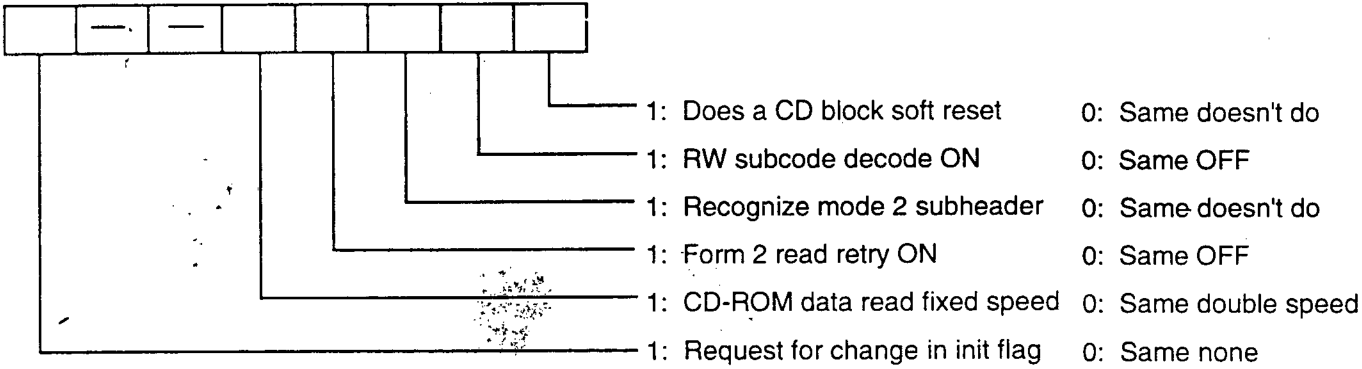
[Function Value] Returns a return code.

[Function]

Initializes the CD block and the CD communications interface. If tray is open, this closes it.

(1) Initial Flag

Set flag for the CD block (8 bit). Default for all bits is set to 0.



(2) Standby Time

Time to go from <PAUSE> to <STANDBY>.

If the standby time passes while in the <PAUSE> state, it is regarded as <STANDBY>.

Set Value	Contents
0x0000	Default (180 seconds).
0x0001 to 0xfffe	Transition time (in seconds).
0xffff	Do not change settings.

(3) ECC Repetitions

Number of times ECC processing (once for each PQ) is repeated when decoding CD-ROM.

Set Value	Contents
0x00	ECC is processed only one time in real time (default value).
0x01 to 0x05	Maximum repeats when there is an error (total 2~6 times).
0x80	Absolutely no ECC processing.
0xff	Do not change settings.

Title	Function	Function Name [SR]	No
Function Specification	Initialize CD Block	CDC_CdInit	1.6

(4) Retries

Number of times to retry to read the same sector when decoding CD-ROM.

Set Value	Contents
0x00	Default (no limit).
0x01 to 0x0f	Repetitions (Stop output after repetitions. Data error.)
0x41 to 0x4f	Repetitions (Data output even if there is an error.) Repetitions are set to lower 4 bits (0 to 15 times).
0x80	Will not retry read.
0xff	Do not change settings.

Title	Function	Function Name [SR]	No
Function Specification	Open Tray	CDC_OpenTray	1.7

[Format] CdcRet CDC_OpenTray(void)
[Input] None
[Output] None
[Function Value] Returns a return code.
[Function]

Opens the tray.

[Comments]

Function can be called for either front-loading or top-loading CD players.

Title	Function	Function Name [--]	No
Function Specification	Data Ready to Transfer	CDC_DataReady	1.8

[Format] CdcRet CDC_DataReady(void)
[Input] None
[Output] None
[Function Value] Returns a return code.
[Function]
Waits until data transfer preparation is complete.

Title	Function	Function Name [S-]	No
Function Specification	End Data Transfer	CDC_DataEnd	1.9

[Format] CdcRet CDC_DataEnd(Sint32 *txwsiz)
[Input] None
[Output] txwsiz : CD block side data transfer set size (word unit)
[Function Value] Returns a return code.
[Function]
Indicates data transfer end for the CD block.
By comparing the host side data transfer size and txwsiz, you can check for transfer errors if the two sizes don't match.
[Comments]
txwsiz returns the following values.

Value	Explanation
0	Data transfer was interrupted.
CDC_DATAERR_SIZ	Data transfer error occurred.
Other	CD block side data transfer size (word units).

[Example]
To transfer 1 sector (=1024 words), txwsiz returns the following values based on the host side data transfer word count until CDC_DataEnd is executed.

Host Side Transfer Word Count	txwsiz value	Explanation
<1024	0	Data transfer was interrupted.
=1024	1024	All indicated sector data was transferred.
>1024	1024	When transferring over the indicated word count. Dummy data is transferred for the excess amount.

8.2.2 CD Drive

Title	Function	Function Name [SR]	No
Function Specification	Play CD	CDC_CdPlay	2.1

[Format] CdcRet CDC_CdPlay(CdcPly *ply)
[Input] ply : Play parameter
[Output] None
[Function Value] Returns a return code.
[Function]
Plays the CD according the indicated play information.
[Example]

(1) Frame Address Designation

```
CdcPly      ply;

CDC_PLY_PMODE(&ply)        = Play mode;
CDC_PLY_STYPE(&ply)        = CDC_PTYPE_FAD;
CDC_PLY_SFAD(&ply)         = Play start frame address;
CDC_PLY_ETYPE(&ply)        = CDC_PTYPE_FAD;
CDC_PLY_EFAD(&ply)         = Play sector count;

ret   =  CDC_CdPlay(&ply);     /*Play Start */
```

(2) Track Designation

```
CDC_PLY_PMODE(&ply)        = Play mode;
CDC_PLY_STYPE(&ply)        = CDC_PTYPE_TNO;
CDC_PLY_STNO(&ply)         = Play start track number;
CDC_PLY_SIDX(&ply)         = Play start index number;
CDC_PLY_ETYPE(&ply)        = CDC_PTYPE_TNO;
CDC_PLY_ETNO(&ply)         = Play end track number;
CDC_PLY_EIDX(&ply)         = Play end index number;

ret   =  CDC_CdPlay(&ply);     /*Play Start */
```

(3) Designating Abbreviated Values

```
CDC_PLY_STYPE(&ply)     = CDC_PTYPE_DEFAULT; /*Start position is disc start */
CDC_PLY_ETYPE(&ply)     = CDC_PTYPE_DEFAULT; /*End position is disc end */
```

(4) Designating Resume Play
(No change to the play range, start from current pickup position.)

```
CDC_PLY_PMODE(&ply)        = CDC_PMODE_NOCHG;
CDC_PLY_STYPE(&ply)        = CDC_PTYPE_NOCHG;
CDC_PLY_ETYPE(&ply)        = CDC_PTYPE_NOCHG;
```

Title Function Specification	Function Seek Play Position	Function Name [SR] CDC_CdSeek	No 2.2

[Format] CdcRet CDC_CdSeek(CdcPos *pos)
[Input] pos : Position Parameters
[Output] None
[Function Value] Returns a return code.
[Function]
Follows the designated position parameters and seeks the play position.
[Example]
(1) Frame Address Designation

```
CdcPos      pos;

CDC_POS_PTYPE(&pos)        = CDC_PTYPE_FAD;
CDC_POS_FAD(&pos)          = Frame address;

ret   =  CDC_SeekCd(&pos);
```

(2) Track Designation

```
CDC_POS_PTYPE(&pos)        = CDC_PTYPE_TNO;
CDC_POS_TNO(&pos)          = Track number;
CDC_POS_IDX(&pos)          = Index number;

ret   =  CDC_SeekCd(&pos);
```

(3) Stopping CD Play (Designating Abbreviated Values)

```
CDC_POS_PTYPE(&pos)    =  CDC_PTYPE_DEFAULT;

ret   =  CDC_SeekCd(&pos);
```

(4) Pausing CD Play (Designating No Change)

```
CDC_POS_PTYPE(&pos)        = CDC_PTYPE_NOCHG;

ret   =  CDC_SeekCd(&pos);
```

Title Function Specification	Function Fast Forward Play	Function Name [SR] CDC_CdScan	No 2.3

[Format] CdcRet CDC_CdScan(Sint32 direction)
[Input] direction : Scan direction
[Output] None
[Function Value] Returns a return code.
[Function]
CD fast forward play (scans either the forward direction or the reverse direction). Continues scan until a CD play-related function (CD play, seek, stop, pause, resume) is called up.

Direction can be set to the following values.

Value	Explanation
CDC_SCAN_FWD	Fast forward scan
CDC_SCAN_RVS	Reverse scan

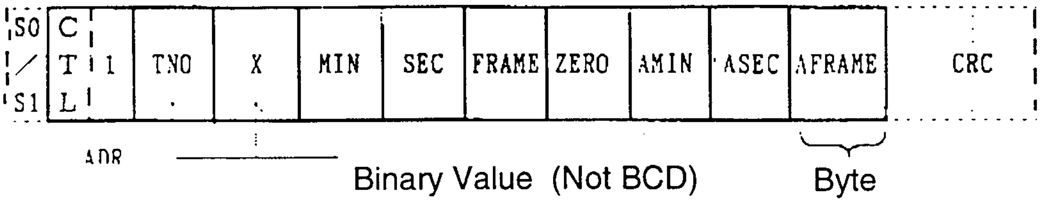
[Comments]
When scanning forward on a CD-ROM, the host must determine, based on the status, that it is a CD-ROM and apply mute to the sound track. (The CD block side can only do digital mute.)

8.2.3 Subcode

Title Function Specification	Function Get Subcode Q	Function Name [S-] CDC_GetScdQch	No 3.1

[Format] CdcRet CDC_GetScdQch(Uint8 *qcode)
[Input] None
[Output] qcode : Range of the subcode Q (10 bytes minus the CRC area)
[Function Value] Returns a return code.
[Function]
Gets the subcode Q.
[Comments]

- If ADR = 1: Track/index numbers are binary values, as shown in the following table.



SO / S1	CTL	1	TNO	X	MIN	SEC	FRAME	ZERO	AMIN	ASEC	AFRAME	CRC

- If other: Subcode Q value as is (10 bytes minus the CRC area).

Title Function Specification	Function Get Subode R to W	Function Name [S-] CDC_GetScdRwch	No 3.2

[Format] CdcRet CDC_GetScdRwch(Uint8 *rwcode, Sint32 *scdflag)
[Input] None
[Output] rwcode : Range of the subcode R to W (pack range 24 bytes)
scdflag : Subcode flag
[Function Value] Returns a return code.
[Function]
Gets the subcode R to W.

8.2.4 CD Device

Title	Function	Function Name [SR]	No
Function Specification	Connect CD Device	CDC_CdSetCon	4.1

[Format] CdcRet CDC_CdSetCon(Sint32 filtno)
[Input] filtno : Filter number to be connected (CDC_NUL_SEL means to separate).
[Output] None
[Function Value] Returns a return code.
[Function]
Sets the destination of the CD device being connected.

Title	Function	Function Name [S-]	No
Function Specification	Get connect status of the CD Device	CDC_CdGetCon	4.2

[Format] CdcRet CDC_CdGetCon(Sint32 *filtno)
[Input] None
[Output] filtno : Filter number to be connected (CDC_NUL_SEL not yet connected).
[Function Value] Returns a retum code.
[Function]
Gets the status of the CD device.

Title	Function	Function Name [S-]	No
Function Specification	Get last sector in the buffer partition	CDC_CdGetLastBuf	4.3

[Format] CdcRet CDC_CdGetLastBuf(Sint32 *bufno)
[Input] None
[Output] bufno : Number of the last stored location in the buffer partition.
CDC_NUL_SEL means not stored in buffer partition.
[Function Value] Returns a return code.
[Function]
Gets the number of the buffer partition in the CD buffer that last had a sector stored in it.

8.2.5 Selector

Title Function Specification	Function Set Filter Sector Range	Function Name [SR] CDC_SetFiltRange	No 5.1

[Format] CdcRet CDC_SetFiltRange(Sint32 filtno, Sint32 fad, Sint32 fasnum)
[Input] filtno : Filter number
fad : Start frame address of the sector range (0: indicates disc start)
fasnum : Frame address sector count (0: indicates disc end)
[Output] None
[Function Value] Returns a return code.
[Function]
Sets the sector range for the filter. If you set the start frame address and frame address sector count each with CDC_FAD_NOCHG, they won't change.

Title Function Specification	Function Get Filter Sector Range	Function Name [S-] CDC_GetFiltRange	No 5.2

[Format] CdcRet CDC_GetFiltRange(Sint32 filtno, Sint32 *fad, Sint32 *fasnum)
[Input] filtno : Filter number
[Output] fad : Start frame address of the sector range (0: indicates disc start)
fasnum : Frame address sector count (0: indicates disc end)
[Function Value] Returns a return code.
[Function]
Gets the sector range of the filter.

Title Function Specification	Function Set Filter Subheader Status	Function Name [SR] CDC_SetFiltSubh	No 5.3

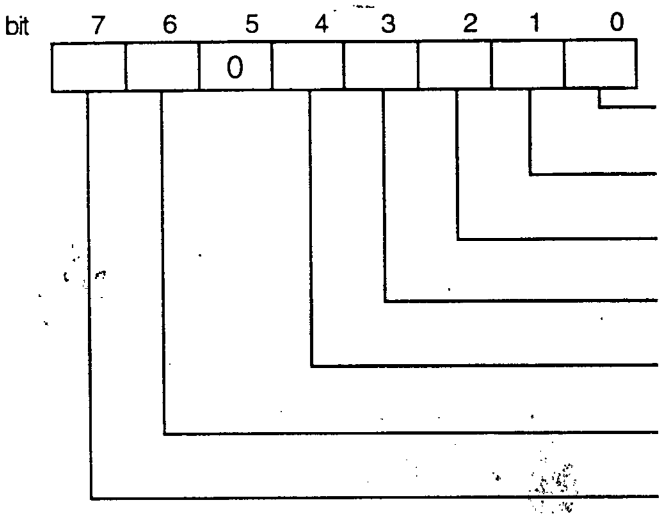
[Format] CdcRet CDC_SetFiltSubh(Sint32 filtno, CdcSubh *subh)
[Input] filtno : Filter number
subh : Subheader status
[Output] None
[Function Value] Returns a return code.
[Function]
Sets the subheader status for the filter.

Title Function Specification	Function Get Filter Subheader Status	Function Name [S-] CDC_GetFiltSubh	No 5.4

[Format] CdcRet CDC_GetFiltSubh(Sint32 filtno, CdcSubh *subh)
[Input] filtno : Filter number
[Output] subh : Subheader status
[Function Value] Returns a return code.
[Function]
Gets the subheader status of the filter.

Title Function Specification	Function Set Filter Mode	Function Name [SR] CDC_SetFiltMode	No 5.5

[Format] CdcRet CDC_SetFiltMode(Sint32 filtno, Sint32 fmode)
[Input] filtno : Filter number
fmode : Filter mode (lower 8 bits are valid)
[Output] None
[Function Value] Returns a return code.
[Function]
Sets the filter mode for the filter.
[Comments]
(1) Filter Mode



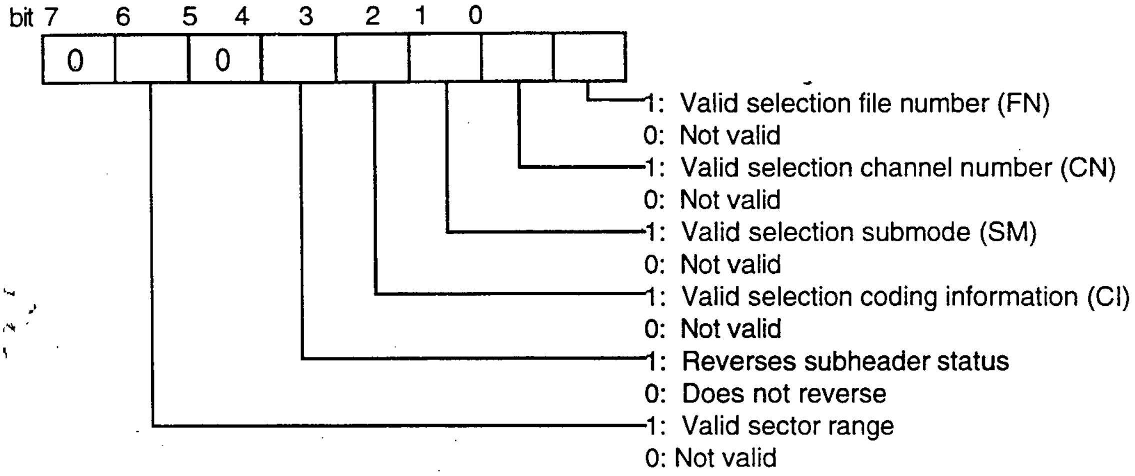
1: Valid selection filenumber (FN)
0: Not valid
1: Valid selection channel number (CN)
0: Not valid
1: Valid selection submode (SM)
0: Not valid
1: Valid selection coding information (CI)
0: Not valid
1: Reverses subheader status
0: Does not reverse
1: Valid sector range
0: Not valid
1: Initialize filter conditions (returns to default value)
0: Not valid

(2) Initializing the Filter Conditions
By setting bit 7 of the filter mode to 1, all other bit settings are ignored and the following default values are used.

- Sector range : Start frame address = 0, sector count = 0.
- Subheader conditions: FN, CN, SMMSK, SMVAL, CIMSK, CIVAL are all 0.
- Filter mode : All bits = 0.

Title Function Specification	Function Get Filter Mode	Function Name [S-] CDC_GetFiltMode	No 5.6

[Format] CdcRet CDC_GetFiltMode(Sint32 filtno, Sint32 *fmode)
[Input] filtno : Filter number
[Output] fmode : Filter mode (lower 8 bits are valid)
[Function Value] Returns a return code.
[Function]
Gets the filter mode of the filter.
[Comments]
Filter Mode


bit 7 6 5 4 3 2 1 0
0
0
1: Valid selection file number (FN)
0: Not valid
1: Valid selection channel number (CN)
0: Not valid
1: Valid selection submode (SM)
0: Not valid
1: Valid selection coding information (CI)
0: Not valid
1: Reverses subheader status
0: Does not reverse
1: Valid sector range
0: Not valid

Title Function Specification	Function Set Filter Connection	Function Name [SR] CDC_SetFiltCon	No 5.7

[Format] CdcRet CDC_SetFiltCon(Sint32 filtno, Sint32 cmode, Sint32 bufno,
Sint32 flnout)

[Input] filtno : Filter number
cmode : Filter connection mode (lower 8 bits are valid)
bufno : Buffer partition number of the true output connector destination (cut off at CDC_NUL_SEL)
flnout : Filter number of the spurious output connector destination (cut off at CDC_NUL_SEL)

[Output] None

[Function Value] Returns a return code.

[Function]

Sets the following connections for the filter.

- True output connector and buffer partition input connector.
- False output connector and other filter input connectors.

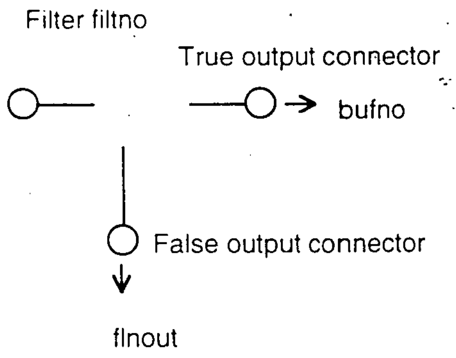
[Comments]

Filter connection mode

bit	7	6	5	4	3	2		
	0	0	0	0	0	0		

True output connector 0: No change to the connection
1: Connect to the buffer partition

False output connector 0: No change to the connection
1: Connect to a filter



Title Function Specification	Function Get Filter Connection	Function Name [S-] CDC_GetFiltCon	No 5.8

[Format] CdcRet CDC_GetFiltCon(Sint32 filtno, Sint32 *bufno, Sint32 *flnout)

[Input] filtno : Filter number

[Output] bufno : Buffer partition number of the true output connector destination (cut off at CDC_NUL_SEL)
flnout : Filter number of the spurious output connector destination (cut off at CDC_NUL_SEL)

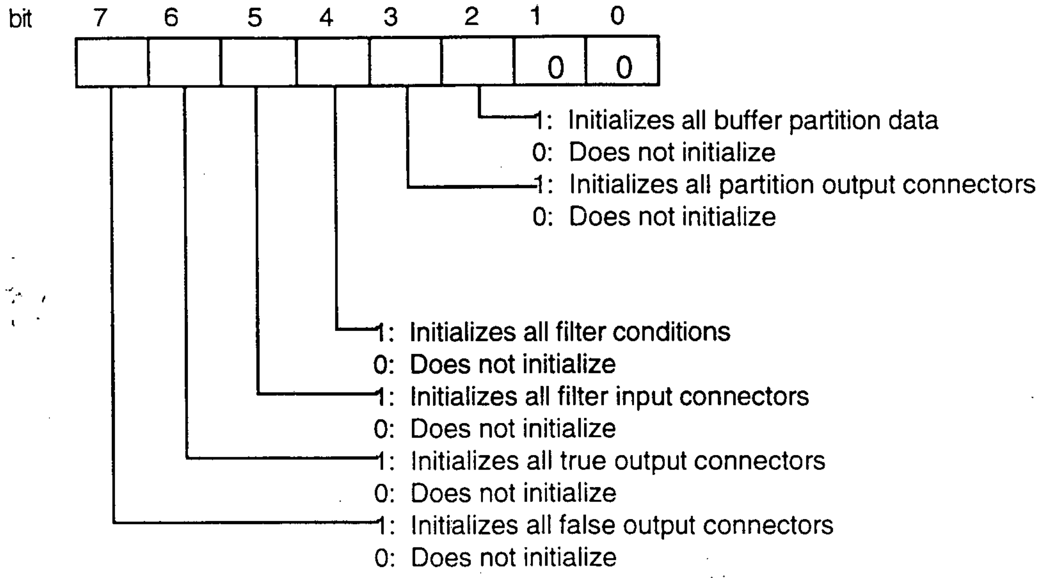
[Function Value] Returns a return code.

[Function]

Gets the filter connection status.

Title Function Specification	Function Reset Selector (Filter, Partition)	Function Name [SR] CDC_ResetSelector	No 5.9

[Format] CdcRet CDC_ResetSelector(Sint32 rmode)
[Input] rmode : Reset mode (lower 8 bits are valid)
[Output] None
[Function Value] Returns a return code.
[Function]
Initializes all selector settings.
[Comments]
(1) Reset Mode



bit	7	6	5	4	3	2	1	0
							0	0

1: Initializes all buffer partition data
0: Does not initialize
1: Initializes all partition output connectors
0: Does not initialize

1: Initializes all filter conditions
0: Does not initialize
1: Initializes all filter input connectors
0: Does not initialize
1: Initializes all true output connectors
0: Does not initialize
1: Initializes all false output connectors
0: Does not initialize

(2) Default Values
Initialization sets the selectors to the following default values.

•	Buffer partition data	: All data is deleted.
•	Partition output connector	: All set to unconnected state.
•	Filter conditions	: Same as initialization with the CDC_SetFiltMode function.
•	Filter input connector	: All set to unconnected state.
•	True output connector	: Partitions and filters with the same numbers are connected.
•	False output connector	: All set to unconnected state.

8.6 Buffer

Title	Function	Function Name [S-]	No
Function Specification	Get CD Buffer Size	CDC_GetBufSiz	6.1

[Format] CdcRet CDC_GetBufSiz(Sint32 *totalsiz, Sint32 *bufnum, Sint32 *freesiz)
[Input] None
[Output] totalsiz : All buffer size (unit: sector)
bufnum : All buffer partition count
freesiz : Empty buffer size (unit: sector)
[Function Value] Returns a return code.
[Function]
Returns the total size and partition count of the CD buffer (both are fixed) and the current empty buffer size.

Title	Function	Function Name [S-]	No
Function Specification	Get Buffer Partition Data Count	CDC_GetSctNum	6.2

[Format] CdcRet CDC_GetSctNum(Sint32 bufno, Sint32 *snum)
[Input] bufno : Buffer partition number
[Output] snum : Sectors in the buffer partition (buffer partition size)
[Function Value] Returns a return code.
[Function]
Gets the sector count (buffer partition size) of the buffer partition.

Title	Function	Function Name [S-]	No
Function Specification	Get Actual Data Size	CDC_GetActSiz	6.3

[Format] CdcRet CDC_GetActSiz(Sint32 bufno, Sint32 spos, Sint32 snum, Sint32 *awsiz)
[Input] bufno : Buffer partition number
spos : Sector position (CDC_SPOS_END: indicates last sector in partition)
snum : Sector count (CDC_SNUM_END: sectors after the indicated sector)
[Output] awsiz : Data size required to get the sector data (word unit)
[Function Value] Returns a return code.
[Function]
Gets the actual data size of the indicated range in the buffer partition.

Title	Function	Function Name [S-]	No
Function Specification	Get Sector Information	CDC_GetSctInfo	6.4

[Format] CdcRet CDC_GetSctInfo(Sint32 bufno, Sint32 spos, CdcSct *sct)
[Input] bufno : Buffer partition number
spos : Sector position
[Output] sct : Sector information
[Function Value] Returns a return code.
[Function]
Gets the header information (FAD) and subheader information from the designated sectors in the buffer partition.

8.2.7 Buffer Input/Output

Title	Function	Function Name [SR]	No
Function Specification	Set Sector Length	CDC_SetSctLen	7.1

[Format] CdcRet CDC_SetSctLen(Sint32 gettype, Sint32 puttype)
[Input] gettype : Sector length type when pulled out
puttype : Sector length type when written
[Output] None
[Function Value] Returns a return code.
[Function]
When taking data from the CD buffer to the host, or when writing data to the CD buffer from the host, this function sets the length of one sector.

The set sector length is reflected when you "delete sector data" or "get actual data size."
[Comments]
Sector Length Type

Constant Name	Explanation
CDC_SCTLEN_2048	2048 bytes (2324 bytes if in mode 2 form 2 and getting data)
CDC_SCTLEN_2336	2336 bytes
CDC_SCTLEN_2340	2340 bytes
CDC_SCTLEN_2352	2352 bytes
CDC_SCTLEN_NOCHG	No change

Title Function Specification	Function Get Sector Data	Function Name [SR] CDC_GetSctData	No 7.2

[Format] CdcRet CDC_GetSctData(Sint32 bufno, Sint32 spos, Sint32 snum)
[Input] bufno : Buffer partition number
spos : Sector position (CDC_SPOS_END: indicates last sector in partition)
snum : Sector count (CDC_SNUM_END: sectors after the indicated sector)
[Output] None
[Function Value] Returns a return code.
[Function]
Acquires the data from the designated sector range in the buffer partition.
[Comments]
After this command is executed, only the actual transfer word count data must be transferred.

Title Function Specification	Function Delete Sector Data	Function Name [SR] CDC_DelSctData	No 7.3

[Format] CdcRet CDC_DelSctData(Sint32 bufno, Sint32 spos, Sint32 snum)
[Input] bufno : Buffer partition number
spos : Sector position (CDC_SPOS_END: indicates last sector in partition)
snum : Sector count (CDC_SNUM_END: sectors after the indicated sector)
[Output] None
[Function Value] Returns a return code.
[Function]
Deletes the data from the designated sector range in the buffer partition.
[Comments]
When you delete sector data, the remaining sector positions all move up.

Title Function Specification	Function Get and Delete Sector Data	Function Name [SR] CDC_GetdelSctData	No 7.4

[Format] CdcRet CDC_GetdelSctData(Sint32 bufno, Sint32 spos, Sint32 snum)
[Input] bufno : Buffer partition number
spos : Sector position (CDC_SPOS_END: indicates last sector in partition)
snum : Sector count (CDC_SNUM_END: sectors after the indicated sector)
[Output] None
[Function Value] Returns a return code.
[Function]
Gets the data from the designated sector range in the buffer partition. Deletes the sector data in the indicated sector range from the buffer partition.
[Comments]
After this command is executed, only actual transfer word count data need be transferred out. At this point even if all data is not transferred and the transfer is ended, the indicated sector range data is deleted.

Title	Function	Function Name [S-]	No
Function Specification	Write Sector Data	CDC_PutSctData	7.5

[Format] CdcRet CDC_PutSctData(Sint32 bufno, Sint32 snum)
[Input] bufno : Buffer partition number to be written
snum : Sector count
[Output] None
[Function Value] Returns a return code.
[Function]
Writes sector data to the buffer partition.
[Comments]
After this command has been executed, only the actual transfer word count should be written.

Title	Function	Function Name [SR]	No
Function Specification	Copy Sector Data	CDC_CopySctData	7.6

[Format] CdcRet CDC_CopySctData(Sint32 srcbn, Sint32 spos, Sint32 snum, Sint32 dstfln)
[Input] srcbn : Buffer partition number to be copied
spos : Sector position (CDC_SPOS_END: indicates last sector in partition)
snum : Sector count (CDC_SNUM_END: sectors after the indicated sector)
dstfln : Copy destination filter number
[Output] None
[Function Value] Returns a return code.
[Function]
Copies data from the indicated sector range in the buffer partition.

Title	Function	Function Name [SR]	No
Function Specification	Move Sector Data	CDC_MoveSctData	7.7

[Format] CdcRet CDC_MoveSctData(Sint32 srcbn, Sint spos, Sint32 snum, Sint32 dstfln)
[Input] srcbn : Buffer partition number to be moved
spos : Sector position (CDC_SPOS_END: indicates last sector in partition)
snum : Sector count (CDC_SNUM_END: sectors after the indicated sector)
dstfln : Move destination filter number
[Output] None
[Function Value] Returns a return code.
[Function]
Moves data from the indicated sector range in the buffer partition.

Title Function Specification	Function Sector data copy / Get transfer error	Function Name [SR] CDC_GetPlacErr	No 7.8

[Format] CdcRet CDC_GetPlacErr(Sint32 *perr)
[Input] None
[Output] perr : Sector data copy/move error information
[Function Value] Returns a return code.
[Function]
Gets the sector data copy/move error information.
[Comments]
Sector data copy/move error information

Constant Name	Explanation
CDC_PERR_OK	Ended normally.
CDC_PERR_NG	Error occurred. • Selector separated during copy/move. • Empty area in the buffer ran out.
CDC_PERR_BUSY	Processing copy/move.

8.2.8 CD Block File System

Title	Function	Function Name [SR]	No
Function Specification	Change Directories	CDC_ChgDir	8.1

[Format] CdcRet CDC_ChgDir(Sint32 fid, Sint32 filtno)
[Input] fid : File identifier of the directory file
CDC_NUL_FID: indicates root directory
filtno : Filter number for the CD block file system
[Output] None
[Function Value] Returns a return code.
[Function]
Changes from the current directory to the directory indicated by the file identifier and reads the file information from the start.

An error results if the indicated file is not a directory.
[Comments]
Filter Number for the CD Block File System: a selector is used in the CD block file system for jobs.

Title	Function	Function Name [SR]	No
Function Specification	Retains File Information	CDC_ReadDir	8.2

[Format] CdcRet CDC_ReadDir(Sint32 fid, Sint32 filtno)
[Input] fid : File identifier of the lead file of the file information group
filtno : Filter number for the CD block file system
[Output] None
[Function Value] Returns a return code.
[Function]
Reads the directory code of the current directory and holds file information for the 254 files that follow the file indicated by the file identifier.

Title	Function	Function Name [S-]	No
Function Specification	Get File Information Range	CDC_GetFileScope	8.3

[Format] CdcRet CDC_GetFileScope(Sint32 *fid, Sint32 *infnum)
[Input] None
[Output] fid : File identifier of the lead file of the held file information range
infnum : Number of information files being held
[Function Value] Returns a return code.
[Function]
Returns the range of file information currently held in the CD block.

Title	Function	Function Name [S-]	No
Function Specification	Get File Information	CDC_GetFileInfo	8.4

[Format] CdcRet CDC_GetFileInfo(Sint32 fid, Sint32 *finfwsiz)

[Input] fid : File identifier (CDC_NUL_FID: all file information)

[Output] finfwsiz : Size of the file information to be transferred (word unit)

[Function Value] Returns a return code.

[Function]

Returns the file information of the indicated file identifier. When CDC_NUL_FID is indicated as the file identifier, this function gets the file information for all of the files in the CD block.

[Comments]

After this function has been executed, only the size of all the file information must be transferred and pulled out.

Title	Function	Function Name [S-]	No
Function Specification	Get Special File Information	CDC_GetOneFileInfo	8.5

[Format] CdcRet CDC_GetOneFileInfo(Sint32 fid, CdcFile *file)

[Input] fid : File identifier

[Output] file : File information

[Function Value] Returns a return code.

[Function]

Returns the file information of the file indicated by the file identifier.

Title	Function	Function Name [SR]	No
Function Specification	Start File Read	CDC_ReadFile	8.6

[Format] CdcRet CDC_ReadFile(Sint32 filtno, Sint32 fid, Sint32 offset)

[Input] filtno : Filter number
fid : File identifier
offset : Offset in sector units

[Output] None

[Function Value] Returns a return code.

[Function]

Starts reading files with respect to the designated filter.

8.2.9 Register Access

Title Function Specification	Function Get Data Transfer Register Pointer	Function Name [- -] CDC_GetDataPtr	No 9.1

[Format] Uint16 *CDC_GetDataPtr(void)
[Input] None
[Output] None
[Function Value]
Returns the data transfer register (DATATRNS) pointer.

Title Function Specification	Function Get Value of Data Transfer Status Register	Function Name [- -] CDC_GetDataStat	No 9.2

[Format] Uint16 CDC_GetDataStat(void)
[Input] None
[Output] None
[Function Value]
Returns the value data status transfer register (DATASTAT).

Title Function Specification	Function Get Value of Interrupt Cause Register	Function Name [- -] CDC_GetHirqReq	No 9.3

[Format] Uint16 CDC_GetHirqReq(void)
[Input] None
[Output] None
[Function Value]
Returns the value of the interrupt cause register (HIRQREQ).

Title Function Specification	Function Clear Interrupt Cause Register	Function Name [- -] CDC_ClrHirqReq	No 9.4

[Format] void CDC_ClrHirqReq(Uint16 bitpat)
[Input] bitpat : Clear bit pattern (cleared bits are 0, others are 1)
[Output] None
[Function Value] None
[Function]
Clears the interrupt cause by clearing the indicated bit pattern (only write 0's).

Title Function Specification	Function Get Value of Interrupt Cause Mask Register	Function Name [- -] CDC_GetHirqMsk	No 9.5

[Format] Uint16 *CDC_GetHirqMsk(void)
[Input] None
[Output] None
[Function Value]
Returns the value of the interrupt cause mask register (HIRQMSK).

Title Function Specification	Function Set Value of Interrupt Cause Mask Register	Function Name [- -] CDC_SetHirqMsk	No 9.6

[Format] void CDC_SetHirqMsk(Uint16 bitpat)
[Input] bitpat : Mask bit pattern (set mask bits to 0, others to 1)
[Output] None
[Function Value] None
[Function]
Masks the interrupt cause through the indicated bit pattern.

Title Function Specification	Function Get MPEG Register Pointer	Function Name [- -] CDC_GetMpegPtr	No 9.7

[Format] Uint16 *CDC_GetMpegPtr(void)
[Input] None
[Output] None
[Function Value]
Returns the MPEG register (MPEGRGB) pointer.

8.2.10 System Functions

Title	Function	Function Name [S -]	No
Function Specification	Confirm CD Block Connection	CDC_SysIsConnect	10.1

[Format] Bool CDC_SysIsConnect(void)
[Input] None
[Output] None
[Function Value] TRUE: CD block is connected.
FALSE: CD block is not connected.

[Function]
Confirms whether the CD block is connected.
Before communications with the CD block start, IPL is called.

Title	Function	Function Name [S -]	No
Function Specification	Direct Issue of CD Block Command	CDC_SysPrimCmd	10.2

[Format] CdcRet CDC_SysPrimCmd(Uint8 *cmdb, Uint8 *rspb)
[Input] cmdb : Command parameters to the CD block (8 bytes)
[Output] rspb : Response from the CD block (8 bytes)
[Function Value] Returns a return code.

[Function]
Issues a command directly to the CD block and gets a response.