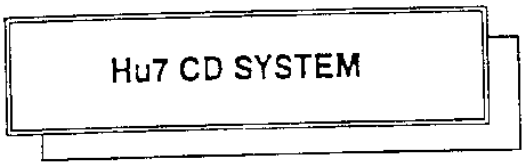


TABLE OF CONTENTS


HARDWARE

Chapter 1 OUTLINE

The Hu7 CD System consists of the hardware and software required to both develope software and create master tapes for the CD-ROM2 system. The Hu7 CD system makes it possible to edit linear PCM and ADPCM data, programs, and data for developing the CD-ROM. It is also possible to create a final master with this system.

1.1 Hu7 CD System Contents

The Hu7 CD system provides an environment for the development of:

. Record editing of linear PCM data
. Record editing of ADPCM data
. Developing programs by CD-ROM2 system emulation
. Registration editing of CD-ROM2 system data
. Creating 8mm MT master
. Data backup to 8mm MT

As shown above, the Hu7 CD system provides a total development environment for the CD-ROM2 system.

Chapter 2 HARDWARE CONFIGURATION

2.1 Hu7 CD System Hardware Configuration

The Hu7 CD system consists of the following hardware:

1) Hu7 CD Contents Hard Disk Unit, 8mm MT unit attached ..... (1)
This is a 620MB hard disk.
Accessories: Power Cable ..... (2)
SCSI Cable ..... (3)

2) Expansion I/O Box ..... (4)
(1) Hu7 CD CDPCM Board (Linear PCM Board) ..... (5)
This board is to be installed in the PC-9801VX interface slot to play linear PCM recordings.
(2) Hu7 CD ADPCM Board ..... (6)
This parallel I/O [illegible]rd is to be installed in the PC-9801VX interface slot to play ADPCM recordings.
Accessories: Parallel Cable ..... (7)

3) SCSI Host Adapter Board ..... (8)
This is a SCSI interface board to install in the PC-9801VX interface slot. It connects to a Hu7 CD contents hard disk unit when used.

Other than the above, the following hardware is required:

1) Hu7 System ..... (9)
PC98 Parallel Interface ..... (10)
Parallel Cable ..... (11)

2) Interface Board for the Hu7 ..... (12)
This interface unit for the CD-ROM2 system is to be installed in the top slot of the Hu7.

3) Display (which is able to input RF/video audio) ..... (13)
Usage : The Hu7 system

4) PC-9801VX + Analog Display + Bus Mouse) ..... (14)
Usage : The CD-ROM2 system emulation

5) PC-9801VX (Memory 640K + possibly VM with 16 color board)
+ Analog Display + Bus Mouse ..... (15)
Usage : The Hu7 emulation

6) Audio System ..... (16)
Usage : Audio check. It would work well with a mixer.

2.2 Hu7 CD System Hardware Connection Method

Now, we are going to explain the connection used by the Hu7 CD system. Numbers listed on the previous page are used. Follow these connection procedures.

1. Connect I/O box (4) to PC-9801 (14)

Digital Connection

2. Install SCSI board (8) to PC-9801 (14)
3. Connect SCSI board (8) and Hu CD contents hard disk (1) with a SCSI cable (3)
4. Install ADPCM board (6) onto the I/O box (4)
5. Install CD system board (12) onto the Hu7 system (9)
6. Connect ADPCM board (6) and CD system board (12) with a parallel cable
7. Install Parallel I/O board (10) on PC-9801 (15)
8. Connect Hu7 system and PC-9801 (15) with a parallel cable (11)

Analog Connection

9. Install CDPCM board (5) onto the I/O box (4)
10. Connect audio "OUT" (16) and CDPCM "IN" (5) with an audio cable
11. Connect CDPCM board "OUT" (5) and CD system board "IN" (12) with an audio cable
12. Connect ADPCM board "IN" (6) and audio "OUT" (16) with an audio cable
13. Connect ADPCM board "OUT" (6) and audio "IN" (16) with an audio cable

Video Connection

14. Connect CD system board "Audio OUT" (12) and display (13) with a video cable

2.3 Individual Hardware

2.3.1 Hu7 CD contents hard disk unit

The Hu7 CD contents hard disk unit is a hard disk with 620MB capacity and performs emulation for the CD-ROM2 system media and drive. Other than emulating the CD-ROM2 using the provided tools, the Hu7 CD contents hard disk unit functions as a MS-DOS hard disk. A 8mm MT accessory is provided with the Hu7 CD contents hard disk unit and it can be used to create a backup master by using software. This Hu7 CD contents hard disk unit also has a IN/OUT bus for SCSI.

2.3.2 Expansion I/O box

This should be installed on the PC-9801VX. The CDPCM board and ADPCM board are installed on this expansion I/O box.

2.3.3 Hu7 CD CDPCM board

This should be installed in the interface slot of the PC-9801VX. It can playback linear PCM recordings using the provided tools. This board has stereo IN/OUT pin jacks. Connecting an audio system to these pins enables creating linear PCM data.

2.3.4 Hu7 ADPCM board

This should be installed in the interface slot of the PC-9801VX. It can playback ADPCM recordings using the provided tools. This board has monaural IN/OUT pin jacks. Connecting an audio system to these pins enables creating ADPCM data. This interface with parallel I/O also transfers data when emulating CD-ROM2.

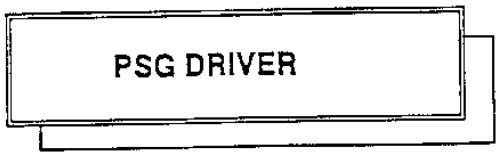
2.3.5 CD system board for Hu7 (Optional)

This should be installed on the top slot of the Hu7. This board has an equivalent function to the CD-ROM2 interface unit IFU-30 and allows CD audio stereo output.

2.3.6 SCSI host adaptor board

This should be installed in the interface slot of the PC-9801VX. This board has a SCSI bus and it should be connected to the Hu7 CD contents hard disk unit.

2.4 Illustration for Hardware Connection


PSG DRIVER

Chapter 1 OUTLINE

This PSG driver contains 12 virtual channels (called tracks) internally ; 6 channels for music (called the main track) and 6 channels for effect (called subtracks), and these are individually controlled. The each track can be separated when necessary, and the work area of the separated track become free.

PSG driver control is made possible with 21 function calls.

With 42 music data commands (internal command level), envelop, vibrato etc., various music data can be created. As the data command levels for the main track and the subtrack are similar, music play of a subtrack is possible.

The PSG driver contains 45 wave form data internally. User-created wave form data and envelop data can be registered outside of the driver.

The PSG driver can be called by either the timer or an IRQ. If the driver is called, the bank where sound data is stored will be spread into a logical address space (2 banks from address $8000), but the user can specify the bank number.

Chapter 2 PSG DRIVER FUNCTION CALL

Note: If following function call is executed, contents of ACC (accumulator) will be destroyed.

2.1 PSG Driver ON (PSG_ON)

_DH=0

Call _AL = 0 Timer Call
1 1RQ Call

Return None

Description Enables the PSG driver. Selects with _AL calling the PSG driver by the timer or by IRQ. If the timer is selected, the timer is set to ON with this function call.

2.2 PSG Driver OFF (PSG_OFF)

_DH=1

Call None

Return None

Description Stops the use of the PSG driver. If timer is selected, it stops the timer operation.

2.3 Initialize PSG (PSG_INIT)

_DH=2

Call _AL = PSG driver system number (0 ~ 5)

Return None

Description Initializes the PSG internal register and work area,then sets the system. The system numbers will set the system as follows:

0 : Enables main track only. Work on sub track side is set free. Stops functions related to subtrack. 1/60 second interval is used for the timer.
1 : Enables subtrack only. Work on main track side is set free. Stops functions related to main track. 1/60 second interval is used for the timer.
2 : Uses both main and sub tracks. 1/60 second interval is used for the timer.
3 : Uses both main and sub tracks. 1/120 second interval is used for the timer.
4 : Uses both main and sub tracks. 1/240 second interval is used for the timer.
5 : Uses both main and sub tracks. 1/300 second interval is used for the timer.

Timer interval settings by system number only effects the subtrack. Main track maintains 1/60 second speed by delay counter. In case of IRQ call interval settings will be ignored.

2.4 Register Sound Data Bank Number (PSG_BANK)

_DH=3

Call _AL = Bank Number 0
_AH = Bank Number 1

Return None

Description Specifies the bank number where sound data is stored. If an interrupt occurs, AL value and AH value will be set to mapping registers MPR4 and MPR5, respectively, and data will be spread into logical address space.

2.5 Register Track Data Index Address (PSG_TRACK)

_DH=4

Call _AX = Top Index Address of Track

Return None

Description Registers the top index address of the track data. (See Chapter 3 "Track Data" for track data format)

2.6 Register Wave Form Data Address (PSG_WAVE)

_DH=5

Call _AX = Wave Form Data Top Address

Return None

Description Registers the top address of user defined wave form data. (See Chapter 5 "Wave Form Data" for wave form data format)

2.7 Register Envelope Data Address (PSG_ENV)

_DH=6

Call _AX = Envelope Data Top Index Address

Return None

Description Registers the top index address of user defined envelope data. (See Chapter 5 "Envelope Data" for envelope data format)

2.8 Register Frequency Modulation Data Address (PSG_FM)

_DH=7

Call _AX = Frequency Modulation Data Top Index Address

Return None

Description Registers the top index address of user defined frequency modulation data. (See Chapter 6 "Frequency Modulation Data" for frequency modulation data format)

2.9 Register Pitch Envelope Data Address (PSG_PE)

_DH=8

Call _AX = Pitch Envelope Data Top Index Address

Return None

Description Registers the top index address of user defined pitch envelope data. (See Chapter 7 "Pitch Envelope Data" for pitch envelope data format)

2.10 Register Percussion Data Address (PSG_PC)

_DH=9

Call _AX = Percussion Data Top Index Address

Return None

Description Registers the top index address of user defined percussion data. (See Chapter 8 "Percussion Data" for percussion data format)

2.11 Set Tempo (PSG_TEMPO)

_DH=10

Call _AL = Tempo Counter (35 ~ 255)

Return None

Description Sets play speed by changing the timer interrupt. If less than 35 is set, it will be corrected to 35. When a sub track is used, try not to change the tempo.

2.12 Play Track Data (PSG_PLAY)

DL=11

Call _AL = Sound Number (0 ~ 127)
_AH = Wave Form Number (effective only at debug mode)

Return None

Description. Plays the track data registered at the specified number. At debug mode, _AH contents will be registered internally as a wave form number. (All tracks to be played are subjected to the change, but if there is specified wave number in the track data, it will be changed to that number.)

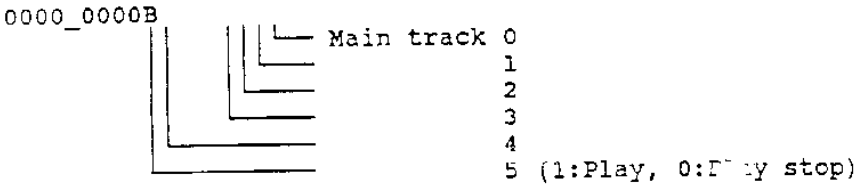
Note: Do not specify number for which track data is not specified.

2.13 Check Main Track (PSG_MSTAT)

_DH=12

Call None

Return ACC = Check Information



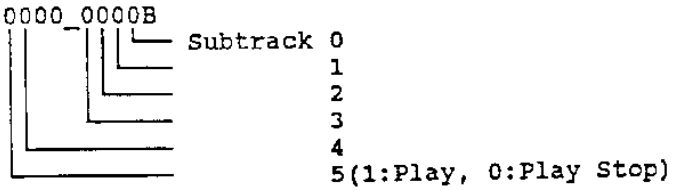
Description Checks the presently playing main track and returns bit information. If a main track is playing, the main track corresponding bit will be set to 1 and the value will be returned. If it is not used, a value $80 will be returned.

2.14 Check Subtrack (PSG_SSTAT)

_DH=13

Call None

Return ACC = Check Information

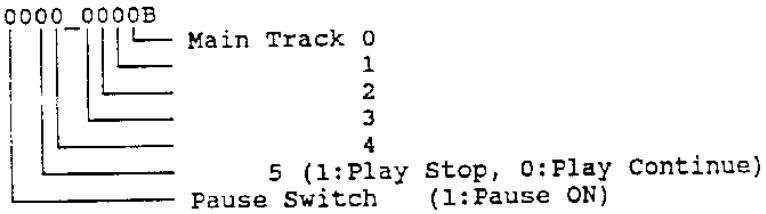


Description Searches the playing subtrack and returns the information as bit information. If it is playing, the corresponding bit to the subtr[illegible]s set to 1 and will b returned. If a subtrack is not used, the value $80 will be returned.

2.15 Stop Main Track Play (PSG_MSTOP)

_DH=14

Call _AL = Specifies Stop Track



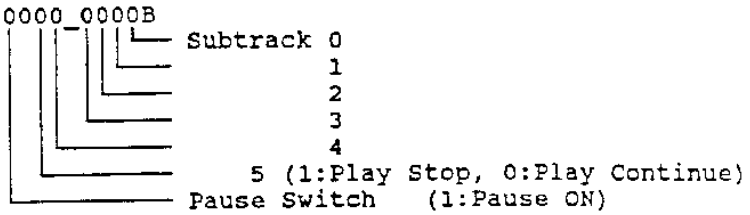
Return None

Description Stops main track play. Specifies stop track with least significant bit 6. If bit 7 is specified, entire main track will be temporarily stopped. (In this case, track specification is ignored.)

2.16 Stop Subtrack Play (PSG_SSTOP)

_DH=15

Call _AL = Specifies Stop Track



Return None

Description Stops subtrack play. Specifies stop track with least significant bit 6. If bit 7 is specified, entire subtrack will be temporarily stopped. (In this case, track specification is ignored.)

2.17 Stop All Track Play (PSG_ASTOP)

_DH=16

Call None

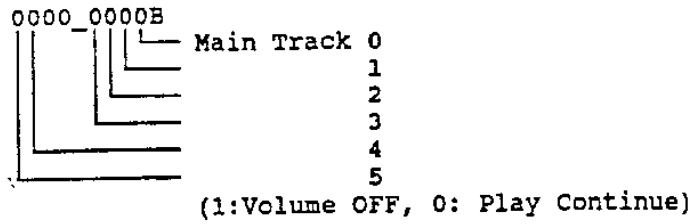
Return None

Description Stops entire track play

2.18 Cut Volume of Main Track (PSG_MVOFF)

_DH=17

Call _AL = Specifies Volume Cut Track



Return None

Description Temporarily turns off the presently playing main track volume. Volume can be returned later.

2.19 Continue Play (PSG_CONT)

_DH = 18

Call _AL = 0 : Restart Main track
1 : Restart Subtrack
2 : Restart All Track

Return None

Description Restarts playing of the track presently stopped or whose volume is off.

2.20 Fade Out (PSG_FDOUT)

_DH = 19

Call _AL = Fade Out Speed (1 ~ 127)

Return None

Description Executes a main track fade out. Fade out speed increases as the value increases. If the specified value is negative, it will be corrected to positive.

2.21 Set Main Track Delay Counter (PSG_DCNT)

_DH = 20

Call _AL = Delay Counter Value (0 ~ 7)

Return None

Description Sets the delay counter. Play speed of the main track slows down as the value increases. Play speed is 1/1 at 0, 1/2 at 1, 1/3 at 2, ... 1/8 at 7.

Note: As the delay counter value is set by PSG initialization, this function is usually not used. Set value to 1 when playing with a tempo change.

Chapter 3 TRACK DATA

3.1 Register Track Data

Registers the top address of data created for each track as shown in the format below and distributes the address to each channel.

```
TRK_INDEX:                    ... Track data index table
     DW      TEST0
     DW      TEST1
          .
          .
          .
;
TEST0:                        ... Track data top address register
table
     DB      0011_1111B
     DW      PART_1           ... Track data top address
     DW      PART_2
     DW      PART_3
     DW      PART_4
     DW      PART_5
     DW      PART_6
;
TEST1:
          .
          .
          .
```

Registers the top address of the track data top address register by using the track data index table. The top address of the index table (TRK_INDEX) is registered to the PSG driver by a function call. In this way, by giving the sound number by a function call, PSG driver will assign the track top address to each channel by the track data top address registration table and will start playing.

The track data top address register table sets how to assign each track data to a channel and if it is registered as a main track or a subtrack. The top 1 byte is a switch and bits 0~5 are switches for channels 1~6. By setting each bit to 1, that channel becomes ON and the track data top address will be assigned from the most recent number within the ON channels.

```
;
TEST1:
     DB      0011_0011B
     DW      PART_A
     DW      PART_B
     DW      PART_C
     DW      PART_D
;
```

In example on left, channel 1,2 and 5,6 are assigned to PART_A,B and PART_C,D respectively. Channel 3,4 becomes OFF.

Bit 7 is an interchange switch for main and sub tracks, and if it is 0, it will be registered as the main track. Bit 6 is a debug switch, and if it is 1, it will be in debug mode.

3.2 Format Track Data

The music track data format used by the PSG driver is shown below. These formats are common for both main and sub tracks except in one area. This driver uses two data creation methods: One is the Tempo method where tempo is set by a change of timer interval. The other is the Length method where play speed is set by adjusting the sound length. With the Tempo method, a change in tempo changes the play speed of the subtrack (effect) as well, therefore, create data using the Length method when using a subtrack.

INTERVAL

2 bytes(Interval code, Sound length) (Direct length mode)
1 byte (Interval code + Sound length) (Time base length mode)

Code				
	Do	: $10	Sol	: $80
	Do#	: $20	Sol#	: $90
	Re	: $30	La	: $A0
	Re#	: $40	La#	: $B0
	Mi	: $50	Ti	: $C0
	Fa	: $60		
	Fa#	: $70	Rest	: $00

Sound length 1 ~ 255 (Direct length mode)
0 ~ 15 (Time base length mode)

Specifies interval. If code is $00 it becomes a rest. There are 2 modes, direct length and time base length, to specify the sound length.

Direct length mode : Directly specifies a sound length between 1~255. It is convenient to specify a sound length directly or to create tempo method data, except the data format is 2 bytes long. The least significant 4 bits in the interval code are ignored.

Time base length mode : Specifies the sound length between 0~15. The specified value is corrected to 1~16 inside the driver. It is convenient to create length method data. Actual length is time bass value times sound length value, which is 1~240. The data format is the most significant 4 bits make up the interval code and the least significant 4 bits make up the sound length data.

Length mode can be interchanged with time bass value.

For tempo method data, the sound length equals 192 divided by the note length. (Quarter note : 192/4=48) At this time, set the delay counter value to 1 with a function call.

For length method data, do not specify the tempo.

TIME BASE : 2 bytes (Time base code, Time base value)

Code $D0

Time base value 0 ~ 15

Specifies the time base value. If value is 0, length mode will become direct mode. If value is other than 0, it will change to time base mode. Default value is 0 (direct length mode).

OCTAVE : 1 byte (Octave code)

Code $D1 ~ $D7

Specifies octave. Octave 1~7 corresponds to code $D1 ~ $D7. If the octave specification is omitted, octave 4 will be specified.

OCTAVE UP : 1 byte (Octave up code)

Code $D8

Elevates one octave up.

OCTAVE DOWN : 1 byte (Octave down code)

Code $D9

Drops one octave down.

TIE : 1 byte (Tie code)

Code $DA

Links front and rear sound together.

TEMPO : 2 bytes (Tempo code, Tempo value)

Code $DB

Tempo value 35 ~ 255

Specifies tempo. If multiple channels are specified, the channel with the most recent number has priority.

• Subtrack does not have this command.

VOLUME : 2 bytes (Volume code, Volume value)

Code $DC

Volume value 0 ~ 31

Specifies volume. Default value is 31.

PAN POT : 2 bytes (Pan pot code, RL value)

Code $DD

RL value $00 ~ $FF

Specifies right and left volume. In the specified value upper is for the left and lower is for the right value. Note: A default value is not defined, so use of this command is necessary.

SOUND LENGTH RATIO : 2 bytes (Sound length ratio, Sound length ratio value)

Code $DE

Sound length ratio value 1 ~ 8

Specifies the sound generation ratio within 1 sound. Ratio between 1/8 ~ 8/8 can be specified by a value 1~8. Default value is 8.

RELATIVE VOLUME : 2 bytes (Relative volume code, Relative value)

Code $DF

Relative value -31 ~ 31

Specifies the volume relatively to the present volume value. Aelative value of 1 raises one volume and -1 brings down one volume.

DAL SEGNO : 1 byte (Dal Segno code)

Code $E1

Returns to the specified location by Segno and repeats play. If there is no segno specified, it will go back to the top of the data. (Da Capo)

SEGNO : 1 byte (Segno code)

Code $E2

Specifies the data location where play will be repeated after using Dal Segno.

REPEAT BEGIN : 2 bytes (Repeat begin code, Loop number)

Code $E3

Loop number 1 - 255

Specifies the beginning point of repeated play and the number of times. If the number of times specified is 0, it will be corrected to 2.

REPEAT END : 1 byte (Repeat end code)

Code $E4

Specifies the ending point of repeated play.

- Repeat begin and repeat end can be nested. One round trip uses 3 bytes of user stack area secured inside. The user stack area secures 1 channel of 12 bytes, therefore, a maximum of 4 levels of nests is possible. More than 4 nests will result in stack overflow, so be careful.
- User stack size for subtracks is 9 bytes.

WAVE : 2 bytes (Wave code, Wave number)

Code $E5

Wave number 0 - 127

Specifies a wave form number. Wave form numbers 0 - 44 are for PSG driver internal definition and 45 - 127 are for user definition. A default is not defined.

Note: Do not specify a number which does not define a wave form.

ENVELOPE : 2 bytes (Envelope code, Envelope number)

Code $E6

Envelope number 0 - 127

Specifies an envelope number. 0 - 15 are for PSG driver internal definition numbers and 16 - 127 are for user definition numbers. Default value is 0.

Note: Do not specify a number which does not define an envelope.

FREQUENCY MODULATION (FM) : 2 bytes (FM code, FM number)

Code $E7

FM number 0 - 127
Specifies a frequency modulation data number.

FM DELAY : 2 bytes (FM delay code, Delay value)

Code $E8

Delay value 0 - 255

Specifies a frequency modulation delay time. Delay value is equivalent to sound length. Default value is 0.

Note: If you have not defined frequency modulation data, set the delay value to 0.

FM CORRECTION : 2 bytes (FM correction code, Standard octave)

Code $E9

Standard octave 0 - 7

Corrects frequency modulation. Standard octave is an octave that has no correcting modulation data. Correction is done by moving an octave up or down. If you specify a standard octave of 0, correction has no effect. Default value is 0. Correction will work on pitch envelope, detune, and sweep, simultaneously.

Note: There is a period where modulation has no effect if an octave is raised too high with correction.

PITCH ENVELOPE : 2 bytes (PE code, PE number)

Code $EA

PE number 0 - 127

Specifies a pitch envelope number.

PE DELAY : 2 bytes (PE delay code, Delay value)

Code $EB

Delay value 0 - 255

Specifies delay time for the pitch envelope. Delay value is equivalent to sound length. Default value is 0.

Note: If you have not defined pitch envelope data, set the delay value to 0.

DETUNE	: 2 bytes (Detune code, Detune value)
Code	$EC
Detune value	-128 ~ 127

Fine tunes sound interval. Default value is 0.

SWEEP	: 2 bytes (Sweep code, Change value)
Code	$ED
Change value	-128 ~ 127

Specifies change value for sweep. The larger the number is, the faster the change becomes. If the change value is positive, the interval goes down. If negative, interval goes up. If 0 is specified, sweep is set to OFF. Default value is 0.

SWEEP TIME	: 2 bytes (Sweep time code, Time value)
Code	$EE
Time value	0 ~ 255

Specifies time for sweep. If 0 is specified, time becomes invalid and will remain in effect until sweep becomes key OFF. Default value is 0.

JUMP	: 3 bytes (Jump code, Address)
Code	$EF
Address	Lower address, Upper address

Jump to track data indicated by the address.

CALL	: 3 bytes (Call code, Address)
Code	$F0
Address	Lower address, Upper address

Calls track data at the address indicated. A return command returns stream back to the data following the call.

RETURN : 1 byte (Return code)

Code $F1

Returns to the data following the call.

- A call and return could be nested. One call uses 2 bytes of user stack. Since the user stack area has 12 bytes, a maximum of 6 levels of nests is possible. Use condition of the user stack needs to be taken into consideration since it is commonly used by the repeat command.
- User stack size for subtracks is 9 bytes.

TRANSPOSE : 2 bytes (Transpose code, Transpose value)

Code $F2

Transpose value -128 ~ 127

Transposes sound interval. A transpose value unit is a half step, and if it is positive, the interval elevates. Default value is 0.

RELATIVE TRANSPOSE : 2 bytes (Relative transpose code, Relative transpose value)

Code $F3

Relative transpose value -128 ~ 127

Relatively transposes from the present transpose value.

ABSOLUTE TRANSPOSE : 2 bytes (Complete transpose code, Transpose value)

Code $F4

Transpose value -128 ~ 127

Transposes all channels. Send this command to the channel with the least recent number.

Note: If transposing an upper area or lower area octave, it might go out of sound range.

VOLUME CHANGE : 2 bytes (V change code, Change quantity)

Code $F5

Change quantity -128 ~ 127

Gradually changes the volume. If the change quantity is positive, the volume will increase and if negative, it will decrease. If 0 is specified, change will be stopped sustaining the present volume. If volume is set while changing, the change will be canceled. Default value is 0.

PAN RIGHT CHANGE : 2 bytes (PR change code, Change quantity)

Code $F6

Change quantity -128 ~ 127

Gradually changes right volume. If change quantity is positive, volume will increase and if negative, it will decrease. If 0 is specified, change will be stopped sustaining the present volume. If pan pot is set while changing, the change will be canceled. Default value is 0.

PAN LEFT CHANGE : 2 bytes (PL change code, Change quantity)

Code $F7

Change quantity -128 ~ 127

Gradually changes left volume. If change quantity is positive, volume will increase and if negative, it will decrease. If 0 is specified, change will be stopped sustaining the present volume. If pan pot is set while changing, the change will be canceled. Default value is 0.

MODE : 2 bytes (Mode code, Mode number)

Code $F8

Mode number 0 - 2

Specifies music play mode. 0 is normal (sound interval) mode, 1 is percussion mode, and 2 is noise mode. 2 is only valid for channel numbers 5 and 6 as the other channels cannot generate this sound correctly. Sound also cannot be generated correctly for mode 1 on channels other than 5 and 6 if noise is contained. Default value is 0.

Note: If mode 2 is specified, interval code $10 becomes noise number 0 and $C0 becomes number 11. There are noise numbers up to 31. Specify noise numbers more than 12 by using transpose.

- Specify rest at percussion mode (mode 1) using one of the following methods:

1. Specify top rest at normal mode (mode 0)

```
PC_PRT:
     DB   $D0,$0F          ; TIME BASE=15
     DB   $DC,$1F          ; VOL=31
     DB   $DD,$EE          ; PAN=$EE
     DB   $F8,$01          ; MODE=1
     DB   $01         ; R
     DB   $01         ; R
     DB   $81         ; G
     DB   $01         ; R
     DB   $81         ; G
     DB   $FF         ; DATA END
```

```
PC_PRT:
     DB   $D0,$0F          ; TIME BASE=15
     DB   $DC,$1F          ; VOL=31
     DB   $DD,$EE          ; PAN=$EE
     DB   $F8,$00          ; MODE=0
     DB   $01         ; R
     DB   $01         ; R
     DB   $F8,$01          ; MODE=1
     DB   $81         ; G
     DB   $01         ; R
     DB   $81         ; G
     DB   $FF         ; DATA END
```

2. Create rest data inside percussion data table

```
--------PERCUSSION DATA TABLE-----------------

PC_INDEX_ADR:
     DW   PC0        ; C
     DW   PC1        ; C+
     DW   PC2        ; D
          :
;
PC0: DB   $F0        ; DATA END   ...  Create this kind of data
;

--------MUSIC DATA----------------------------

PC_PRT:
     DB   $D0,$0F         ; TIME BASE=15
     DB   $DC,$1F         ; VOL=31
     DB   $DD,$EE         ; PAN=$EE
     DB   $F8,$01         ; MODE=1
     DB   $01        ; R
     DB   $01        ; R
     DB   $81        ; G
     DB   $01        ; R
     DB   $81        ; G
     DB   $FF        ; DATA END


PC_PRT:
     DB   $D0,$0F         ; TIME BASE=15
     DB   $DC,$1F         ; VOL=31
     DB   $DD,$EE         ; PAN=$EE
     DB   $F8,$01         ; MODE=1
     DB   $11        ; C (R)
     DB   $11        ; C (R)
     DB   $81        ; G
     DB   $11        ; C (R)
     DB   $81        ; G
     DB   $FF        ; DATA END
-----------------------------------------------
```

FADE OUT : 2 bytes (Fade out code, Speed)

Code $FE

Speed 1 - 127

Performs fade out. The larger the value is, the faster fade out speed is. If the specified value is negative, it will be corrected to positive.

DATA END : 1 byte (Data end code)

Code $FF

Indicates end of track data. Always put this code at a play stop location.

Chapter 4 WAVE FORM DATA

The PSG driver defines 45 kinds of wave form data internally but user created wave form data can be defined outside with a function call. The user can define 83 kinds of wave forms. Wave form numbers 0~44 are for internal definition and 45~127 are for user definition.

One wave form data f... t is 32 bytes of wave form data arranged in memory in order. The top addresses of the data are registered to the PSG driver through a function call.

-------User defined wave form data----------

```
WAVE_TOP:
      DB    $00,$00,$00,$00,$00,$00,$00,$00     ; NO.45
      DB    $1F,$1F,$1F,$1F,$1F,$1F,$1F,$1F
      DB    $00,$00,$00,$00,$00,$00,$00,$00
      DB    $1F,$1F,$1F,$1F,$1F,$1F,$1F,$1F
;
      DB    $00,$01,$02,$03,$04,$05,$06,$07     ; NO.46
      DB    $08,$09,$0A,$0B,$0C,$0D,$0E,$0F
      DB    $10,$11,$12,$13,$14,$15,$16,$17
      DB    $18,$19,$1A,$1B,$1C,$1D,$1E,$1F
;
      DB    $00,$08,$0F,$14,$19,$1B,$1D,$1E     ; NO.47
      DB    $1E,$1D,$1B,$19,$14,$0F,$08,$00
      DB    $1F,$1E,$1D,$1C,$1B,$1A,$09,$08
      DB    $07,$06,$05,$04,$03,$02,$01,$00
;
                                        :
```

Chapter 5 ENVELOPE DATA

In this PSG driver, defined envelope data is not related to wave form data, so the user can select it by number. Envelope data contains 16 definitions inside the driver, but the user can define 112 kinds outside the driver by a function call. Envelope number 0~15 are for internal definition and 16~127 are for user definition.

Envelope data consists of an index and data parts. At registration, register the top address of the index to the PSG driver with a function call.

Envelope data format is as follows:

RELEASE RATE DATA : 3 bytes (Code, Level change quantity)

Code $FB

Change quantity -$7C00 ~ $7C00

Specifies the release rate (change speed after key OFF until level becomes 0). If change quantity is negative, level decreases and if positive, level increases and it will change until the maximum levelis reached.

LEVEL DATA : 3 bytes (Code, Level set value)

Code $FC

Set quantity 0 ~ $7C00

Sets the initial level for time of level change.

DECAY RATE DATA : 3 bytes (Time, Level change quantity)

Time 0 ~ 250

Change quantity -$7C00 ~ $7C00

Specifies the decay rate (level change speed during key ON). If change quantity is positive, level increases and if negative, it decreases. Time sets the time length for level to change during key ON. If time specified is 0, the level will keep changing during key ON. Time value is the equivalent of sound length.

DATA END : 1 byte (Code)

Code $FF

Indicates end of data. If this code is reached during key ON, sustain until the present level changes to key OFF. This does not apply if the decay time is 0.

Level change quantity value $400 is equal to level 1 and the maximum $7C00 is equal to level 31.

Place data in the order of number. (Put the release rate on top otherwise it will be regarded as an omission.)

Release, level, and decay data can be omitted, and data will be set to 0.

Release rate will be automatically set to 0 for envelope data used for percussion data. (Set release data will be ignored.)

Multiple level and decay data can be set. That will allow you to define complicated envelopes. (Data number for one envelope data is up to 85.)

If sound length ratio is 8, release will have no effect.

——User defined envelope data——————

```
ENV_INDEX:
     DW   ENV16
     DW   ENV17
            :
;
ENV16:
     DB   $FB          ; RELEASE RATE DATA
     DW   -$100
     DB   $FC          ; LEVEL DATA
     DW   31*$400
     DB   25           ;DECAY RATE DATA
     DW   -$180
     DB   $FF          ; DATA END
;
ENV17:
     DB   $FB
     DW   -$80
     DB   $FC
     DW   25*$400
     DB   3
     DW   $800
     DB   0
     DW   -$C0
     DB   $FF
;
               :
```

Chapter 6 FREQUENCY MODULATION DATA

This frequency modulation data generates soft LFO for a sound interval and it can generate vibrato effects, etc. One modulation data table consists of one wave form cycle and various wave forms can be created using the data. One byte of data is an additional value for sound interval frequency number of sound length 1.

Frequency modulation data values are -127 ~ 127 and -128($80) indicates end of data. The modulation table length maximum is 256 bytes. Data configuration consists of index and data parts. At registration, register the top address of the index to the PSG driver with a function call.

------User defined frequency modulation data----------

```
FM_INDEX:
      DW    FMD0
      DW    FMD1
      DW    FMD2
              :
;
FMD0:
      DB    00,01,02,03,02,01,00,-1,-2,-3,-2,-1,$80
FMD1:
      DB    00,-2,-4,-6,-4,-2,00,02,04,06,04,02,$80
FMD2:
      DB    03,03,02,02,01,01,00,00,-1,-1,-2,-2,-3,-3,$80
                        :
```

Chapter 7 PITCH ENVELOPE DATA

Pitch envelope data is a data table used to partially change pitch (sound interval). Data configuration is the same as for frequency modulation data.

At registration, register the top address of index part with a function call.

——User defined pitch envelope data——

```
PE_INDEX:
      DW    PEG0
      DW    PEG1
      DW    PEG2
              :
;
PEG0:
      DB    -15,-12,-10,-8,-6,-5,-3,-2,-1,00,$80
PEG1:
      DB    0,1,2,4,6,8,12,15,17,00,$80
PEG2:
      DB    -1,-3,-6,-10,-13,-14,-13,-10,-6,-3,-1,00,$80
                                              :
```

Chapter 8 PERCUSSION DATA

This PSG driver creates percussion data with effective sound and plays the data using interval.

Data configuration is divided into index and data parts. At registration, register the top address of the index to the PSG driver with a function call.

12 addresses can be defined for the index part and sound interval code $10 ~ $C0 will be assigned from its first address. When generating sound from percussion data, define assigned interval code by track data so that the data will play. (When play mode 1) Data part defines percussion data using the following format. Sound will be generated in order of definition.

NOISE NUMBER

$00 ~ $1F

Defines noise number. This data becomes the length of sound length 1.

INTERVAL FREQUENCY NUMBER

$B0 + Upper frequency number data, Lower frequency number data

Defines the interval frequency number. This data becomes the length of sound length 1.

ENVELOPE NUMBER

$C0, 0 ~ 127

Specifies the envelope number. The release rate of envelope data specified here will be set to 0.

PAN POT

$D0, $00 ~ $FF

Specifies pan pot. Upper data is for the left and lower is for right specification.

WAVE FORM NUMBER

$E0, 0 ~ 127

Specifies the wave form number.

DATA END

$F0

Indicates end of data.

——User defined percussion data——

```
PC_INDEX:
      DW    PCN0            ; C  ($10)
      DW    PCN1            ; C+ ($20)
      DW    PCN2            ; D  ($30)
      DW    PCN3            ; D+ ($40)
      DW    PCN4            ; E  ($50)
      DW    PCN5            ; F  ($60)
      DW    PCN6            ; F+ ($70)
      DW    PCN7            ; G  ($80)
      DW    PCN8            ; G+ ($90)
      DW    PCN9            ; A  ($A0)
      DW    PCN10           ; A+ ($B0)
      DW    PCN11           ; B  ($C0)
;
PCN0:
      DB    $C0,7           ; ENVELOPE NO.
      DB    $E0,6           ; WAVE NO.
      DB    $D0,$FF         ; PAN POT
      DB    $B0+3,$AC       ; TONE FRQ
      DB    $B0+4,$0C
      DB    $1F             ; NOISE FRQ
      DB    $1B
      DB    $1D
      DB    $1A
      DB    $F0             ; DATA END
;
PCN1:
      DB    $C0,21
      DB    $E0,6
      DB    $D0,$FF
      DB    $B0+4,$7C
      DB    $B0+4,$8C
      DB    $B0+4,$9C
      DB    $B0+4,$BC
      DB    $B0+4,$DC
      DB    $B0+4,$FC
      DB    $B0+5,$0C
      DB    $F0
;
            :
            :
```

CD-ROM SYSTEM
BIOS Ver1.00

TABLE OF CONTENTS

```
;
;       IPL INFORMATION BLOCK DATA FORMAT (DATA RECORD TOP + 1)
;
        db      ?       ;00 IPLBLK H    ;load start record no. of CD
        db      ?       ;01 IPLBLK M    ;load start record no. of CD
        db      ?       ;02 IPLBLK L    ;load start record no. of CD
        db      ?       ;03 IPLBLN      ;load block length of CD
        db      ?       ;04 IPLSTA L    ;program load address L
        db      ?       ;05 IPLSTA H    ;program load address H
        db      ?       ;06 IPLJMP L    ;program execute address L
        db      ?       ;07 IPLJMP H    ;program execute address H
;
        db      ?       ;08 IPLMPR2     ;ipl set mpr2 (+ max_mapping)
        db      ?       ;09 IPLMPR3     ;ipl set mpr3 (+ max_mapping)
        db      ?       ;10 IPLMPR4     ;ipl set mpr4 (+ max_mapping)
        db      ?       ;11 IPLMPR5     ;ipl set mpr5 (+ max_mapping)
        db      ?       ;12 IPLMPR6     ;ipl set mpr6 (+ max_mapping)
;
        db      ?       ;13 OPENMODE    ;opening mode
                                        ;bit7654_3210
                                        ;       |||└─ data read to vram
                                        ;       |||   0:not read, 1:read
                                        ;       ||└── data read to adpcm buffer
                                        ;       ||    0:not read, 1:read
                                        ;       |└─── bg display
                                        ;       |     0:display on, 1:display off
                                        ;       └──── adpcm play
                                        ;             0:play, 1:not play
                                        ;             adpcm play mode
                                        ;             0:single, 1:repeat
;
        db      ?       ;14 GRPBLK H    ;opening graphic data record no.
        db      ?       ;15 GRPBLK M    ;opening graphic data record no.
        db      ?       ;16 GRPBLK L    ;opening graphic data record no.
        db      ?       ;17 GRPBLN      ;opening graphic data length
        db      ?       ;18 GRPADR L    ;opening graphic data read address L
        db      ?       ;19 GRPADR H    ;opening graphic data read address H
;
        db      ?       ;20 ADPBLK H    ;opening ADPCM data record no.
        db      ?       ;21 ADPBLK M    ;opening ADPCM data record no.
        db      ?       ;22 ADPBLK L    ;opening ADPCM data record no.
        db      ?       ;23 ADPLEN      ;opening ADPCM data length
        db      ?       ;24 ADPRATE     ;opening ADPCM sampling rate
;
        db      0       ;25             ;(reserve)
        db      0       ;26             ;(reserve)
        db      0       ;27             ;(reserve)
        db      0       ;28             ;(reserve)
        db      0       ;29             ;(reserve)
        db      0       ;30             ;(reserve)
        db      0       ;31             ;(reserve)
;
        db      'PC Engine CD-ROM SYSTEM',0     ;(ID string)
        db      'Copyright HUDSON SOFT / NEC Home Electronics, Ltd.',0
        db      '               '               ;program name     (16 bytes)
        db      '      '                        ;                 ( 6 bytes)
;
```

IPL INFORMATION

IPLBLK	Top record no. where the program is contained
IPLBLN	No. of records for program to read
IPLSTA	Main memory address for program read
IPLJMP	Starting address of execution after program read
IPLMPR	Bank no. to set to MPR before program read
OPENMODE	Opening mode

By setting the bits (bit5,0) to the value specified by OPENMODE in IPL INFORMATION, data can be loaded into VRAM and displayed before loading the user program.

```
bit 7654_3210
      |     |
      |     └── 0: Do not load
      |         1: Load
      └──────── 0: Display ON
                1: Display OFF
```

GRPBLK H, GRPBLK M, and GRPBLK L specifies the top record of data to load.

GRPADR L and GRPADR H specifies the top VRAM address into which BG font data is read.

GRPBLN specifies the total record that contains color palette data, BAT data, and BG font data in the format shown below.

```
                IPL GRAPHIC DATA FORMAT
GRPBLK     ┌───────────────────┐.........................
           │ COLOR PALETTE     │       1 RECORD        :
           ├───────────────────┤.......................:
           │ BAT DATA          │       1 RECORD        :
           ├───────────────────┤.................    GRPBLN
           │                   │                       :
           │ BG FONT DATA      │ GLPBLN - 2 RECORD     :
           │                   │                       :
           └───────────────────┘.......................:...
```

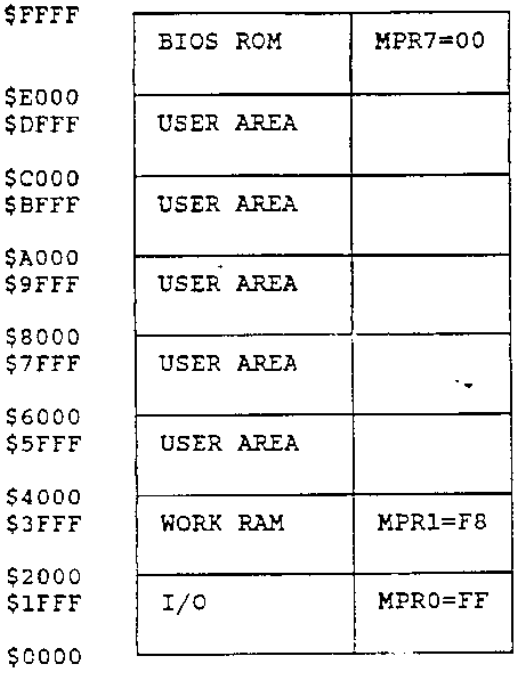
By setting the bits (bit7,6,1) to the value specified by OPENMODE in IPL INFORMATION, data can be loaded into ADPCM buffer and displayed before loading the user program.

```
bit 7654_3210
    ||      |
    ||      └── 0: Load
    ||          1: Do not load
    |└───────── 0: Playback
    |           1: Do not playback
    └────────── 0: Playback once
                1: Repeat playback
```

ADPBLK H, ADPBLK M, and ADPBLK L specifies the top record of data to load.
ADPBLN specifies the number of ADPCM data record.
ADPRATE specifies the ADPCM sampling rate.

Normal Memory Layout

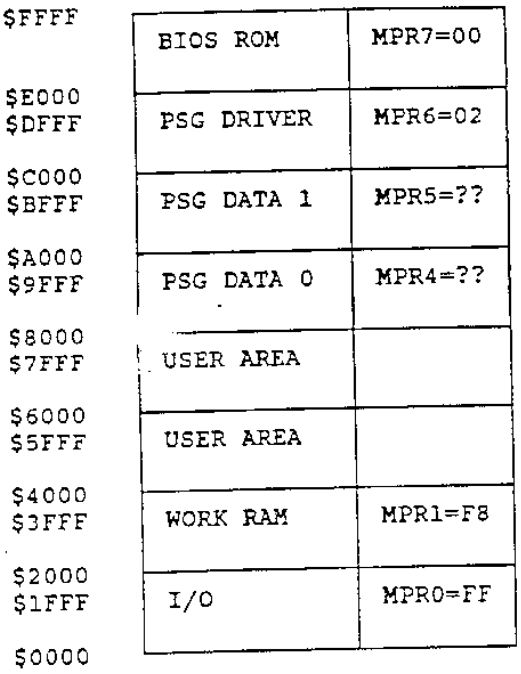
Logical Address



When calling BIOS or using an interrupt routine from BIOS, MPR0,1,7 cannot be changed.

Memory Layout At PSG Driver Load Time

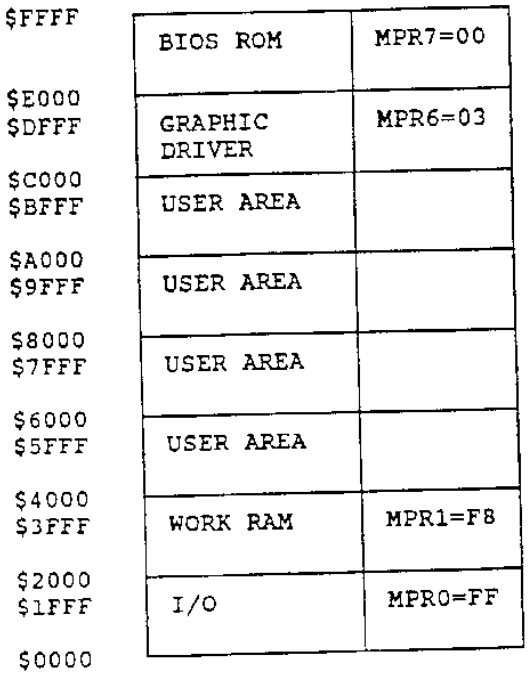
Logical Address



When the PSG driver is loaded, MPR 4,5,6 are set as shown at left. After completion of the PSG driver, they will be reset to the user set status. A user interrupt routine cannot be set here.

Memory Layout At Graphic Driver Load Time

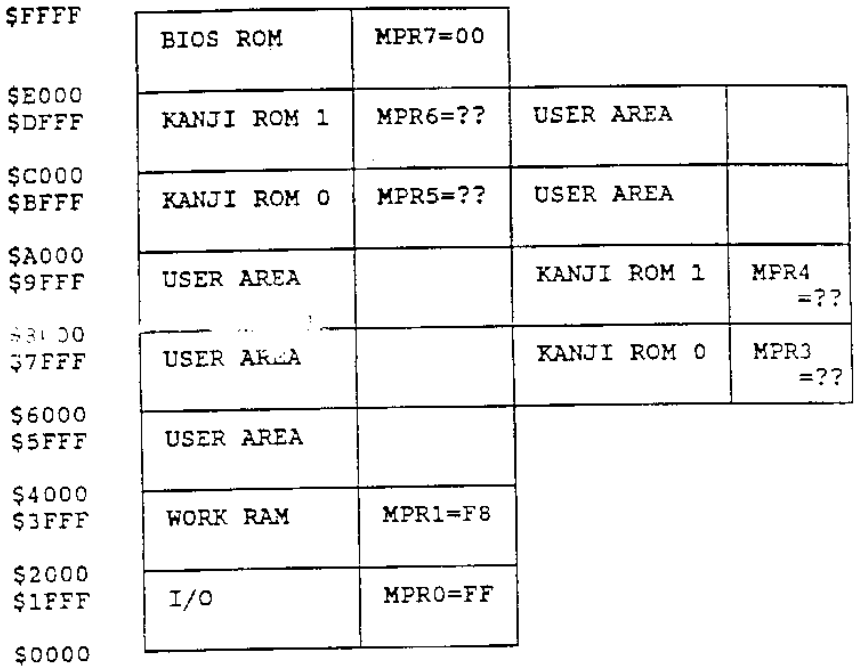
Logical Address



When GRAPHIC DRIVER is loaded, MPR 6 is set as shown at left. After completion of the GRAPHIC DRIVER, it will be reset to the user set status. A user interrupt routine cannot be set here.

Memory Layout At Kanji ROM Load Time

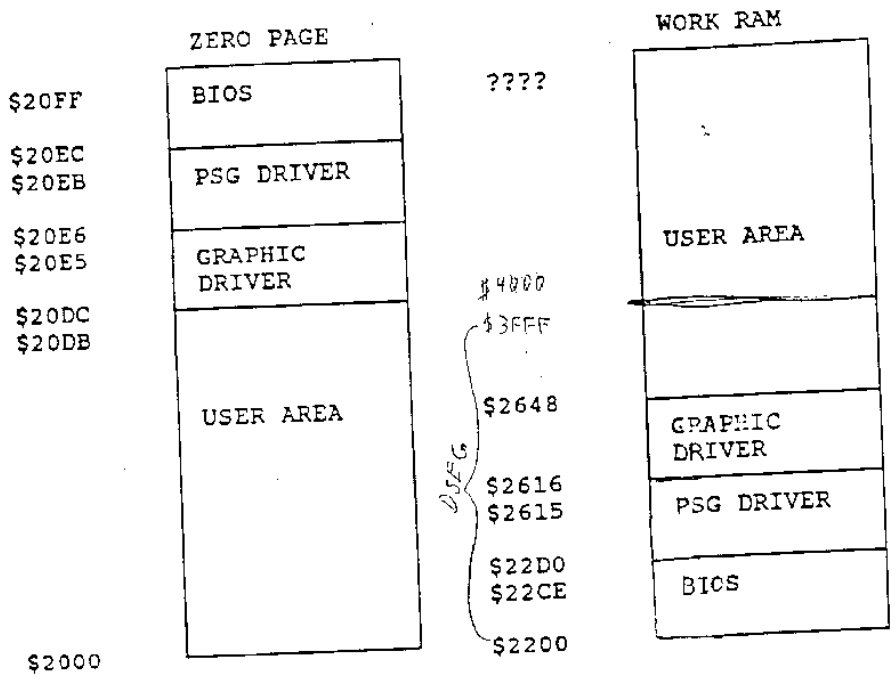
Logical Address



Two kinds of mapping shown above are possible. Which map is used depends on the specified Kanji font data destination address.

Destination address: below $A000, Kanji ROM $A000 - $DFFF
Destination address: above $A000, Kanji ROM $6000 - $9FFF

RAM Memory Layout



Working area of the PSG DRIVER and the GRAPHIC DRIVER can be used as a user area when each driver is not used. Do not destroy the BIOS area.

[$00] CD_BOOT: Boot CD-ROM BIOS

IN None
OUT None

Description Shifts to CD-ROM BIOS boot status.
Displays the title picture and waits for the RUN command.

[$01] CD_RESET: Reset CD drive

IN None
OUT AREG: Execution result
$00 OK
ELSE SUB ERROR CODE
(See SUB ERROR CODE list)

Description Brings CD-ROM drive to the power up state.
If drive is reset, the drive will check DISC and read TOC when next command is issued. It will take a few seconds to reach ready status, therefore, make sure to check CD_STAT to see if the drive is ready before issuing the next command. Do not use this call except for a DISC change. (Not necessary to call this command for initialization of application.)

[$02] CD_BASE: Set Base Address of Data Record No.

IN

_BH:	Base address type %0000_00XX	%0100_00XX	%1000_00XX
	LOG.BLOCK	CD-ATIME	CD-TNO
_AL:	A_REC H	AMIN(BCD)	TNO(BCD)
_AH:	A_REC M	ASEC(BCD)	NO USE
_BL:	A_REC L	AFRAME(BCD)	NO USE

_CL: Set mode
bit 7 6 5 4 3 2 1 0
XXXXXX00 Set both
XXXXXX01 Set first
XXXXXX10 Set second
XXXXXX11 Do not set

OUT _AL: A_REC H
_AH: A_REC M
_BL: A_REC L

Description There are 2 sets of base addresses. Normally, the system sets the head record (IPL RECORD) of the top data track to both addresses during IPL. At data read, based on the base address the user will specify an offset read address. Normally the first set base address is used but when a not recoverable DISC damage occurs, the second set is used. Specify the top of the alternate track prepared by the user for errors as the second base address. The first base address is changeable. If an alternate track is not used, use the same address for the first second base address.

[$03] CD_READ: Read Data from CD

IN

```
_CL: REC H
_CH: REC M
_DL: REC L

_DH: Data read address type
     01:LOCAL   FE:VRAM     FF:VRAM    2~6:MPR NO.
_BL: ADR L      ADR L       ADR L      BANK NO. ($80~$87)
_BH: ADR H      ADR H       ADR H      NO USE

_AL: REC LEN    BYTE LEN L  REC LEN    REC LENGTH
_AH: NO USE     BYTE LEN H  NO USE     NO USE
```

OUT

```
AREG:     Execution result
          $00        OK
          ELSE       SUB ERROR CODE
```

Description Length of a record is 2KB. VDTIN_FLG is set to 1 during a read to VRAM. Normal operation is not guaranteed if MAWR or MARR is changed by interrupt, etc. during this time. Read using an MPR NO. will read to the specified RAM BANK using the specified MPR.NO as an access window. Example, MPR=3, $6000 ~ $7FFF If the number of record is more than 4 (8KB), the specified bank number will be changed by incrementing one. The current bank number will be saved and restored.

[$04] CD_SEEK: Seek CD to a Specified Address

IN

_CL: REC H
_CH: REC M
_DL: REC L

OUT None

Description Moves the head of the CD-ROM to a specified address. The head will automatically be returned to the home position and the drive motor will be turned off if no commands are received for 15 minutes. A command issued after this will take longer to complete as the motor must be restarted and the head must be moved back into position.

[$05] CD_EXEC: Read Data From CD to Specified Address

IN

_CL: REC H
_CH: REC M
_DL: REC L

_DH: Data read address type
01:LOCAL ELSE:MPR NO. (2 ~ 6)
_BL: ADR L BANK NO. ($80 ~ $87)
_BH: ADR H NO USE

_AL: REC LENGTH REC LENGTH
_AH: NO USE NO USE

Description It will be reset if an error occurs.

[$06] CD_PLAY: Search and Playback CD Audio

```
IN   _BH: Playback start address type
          $0000_0000     $0100_0000     $1000_0000     $1100_0000
          LOG.BROCK      CD-ATIME       CD-TNO         CURRENT
     _AL: A_REC H        AMIN(BCD)      TNO(BCD)       NO USE
     _AH: A_REC M        ASEC(BCD)      NO USE         NO USE
     _BL: A_REC L        AFRAME(BCD)    NO USE         NO USE

     _DH: Playback end address type
          $00YY_YYYY     $01YY_YYYY     $10YY_YYYY     $11YY_YYYY
          LOG.BROCK      CD-ATIME       CD-TNO         READ OUT
     _CL: A_REC H        AMIN(BCD)      TNO(BCD)       NO USE
     _CH: A_REC M        ASEC(BCD)      NO USE         NO USE
     _DL: A_REC L        AFRAME(BCD)    NO USE         NO USE

     _DH: Play mode
          bit 7654_3210
                YY_YYYY
              XX00_0000  Mute play.  Status will return right
                         after command is received.

              XX00_0001  Infinite repeat play.  Drive is busy
                         after command is received until the next
                         command issuance.

              XX00_0010  Normal play.  Drive is busy after the
                         command is received until playback
                         completion.

              XX00_0011  Normal play.  Status will return right
                         after the command is received.

              XX00_0100  Do not change from previously set mode.

OUT  AREG:     Execution result
               $00  OK
               ELSE SUB ERROR CODE
```

Description After fade out execution, and if not canceled, the sound circuit will be set to muting status. Cancel fade out by issuing CD_FADE in advance.

[$07] CD_SEARCH: Search CD Audio Track

```
IN    _BH: Address type of search objective
           %0000_00XX       %0100_00XX      %1000_00XX
           LOG.BROCK        CD-ATIME        CD-TNO
      _AL: A_REC H          AMIN(BCD)       TNO(BCD)
      _AH: A_REC M          ASEC(BCD)       NO USE
      _BL: A_REC L          AFRAME(BCD)     NO USE
      _BH: Search mode
           bit 7654_3210
                XXXX_XX??
                       ||
                       |└── 0 : No return until search is completed
                       |    1 : Return after search without
                       |        completing the search.
                       └─── 0 : Pause where search is completed.
                            1 : Play from where search is
                                completed. (To DI...)
```

```
OUT   AREG:     Execution result
                $00   OK
                ELSE SUB ERROR CODE
```

Description When search mode is set to 1, the execution result will always be 0. To find errors, issue CD-STAT set to drive BUSY check. After checking, SCSISTS will be set to 0 if normal termination occurred, and to a non-zero number if an error occurred. (The search operation and the contents of SCSISTS will not be guaranteed if a drive command is issued before the search command was completed.)

[$C] CD_PAUSE: Pause CD Audio Playback

IN None

OUT AREG: Execution result
$00 OK
ELSE SUB ERROR CODE

Description Stops playback at the current position during audio playback. In order to continue playback, call CD_PLAY using the current position as the playback starting address.

[$09] CD_STAT: Check CD Drive Status

IN AREG: Get status mode
$00 Drive busy check
ELSE Drive unit ready check

OUT AREG: Execution result

BUSY CHECK	READY CHECK
$00 NOT BUSY	$00 READY
ELSE BUSY	ELSE SUB ERROR CODE

[$0A] CD_SUBQ: Read Play Status and Sub Code Q

```
IN   _BX:        Read buffer (10 bytes) address

OUT  AREG:       Execution result
                 $00         OK
                 ELSE        SUB ERROR CODE
Description      SUBCODE-Q 10bytes
                 0: PLAYING STATUS
                           $00 PLAYING
                           $01 STILL
                           $02 PAUSE
                           $03 NOT PLAYING

                 1: CONTROL/ADDRESS
                 bit 3 2 1 0
                     0 0 X 0  2 AUDIO CHANNELS WITHOUT PRE-EMPHASIS
                     0 0 X 1  2 AUDIO CHANNELS WITH PRE-EMPHASIS
                     1 0 X 0  4 AUDIO CHANNELS WITHOUT PRE-EMPHASIS
                     1 0 X 1  4 AUDIO CHANNELS WITH PRE-EMPHASIS
                     0 1 X 0  DATA TRACK
                     0 1 X 1  RESERVED
                     1 1 X X  RESERVED
                     X X 0 X  DIGITAL COPY PROHIBITED
                     X X 1 X  DIGITAL COPY PERMITTED

                 2: TRACK NO(BCD)
                 3: INDEX(BCD)
                 4: REAL MIN(BCD)
                 5: REAL SEC(BCD)
                 6: REAL FRAME(BCD)
                 7: TOTAL MIN(BCD)
                 8: TOTAL SEC(BCD)
                 9: TOTAL FRAME(BCD)
```

Data contents are guaranteed if the drive is ready and the disk is rotating.

[$0B] CD_DINFO: Read TOC (Table Of Contents) Data.

IN _BX: Read buffer (4 bytes) address

_AL: Read data type

%0000_0000	%0000_0001	%0000_0010	%0000_0011
MIN TNO(BCD)	READOUT AMIN	AMIN(BCD)	A_REC H
MAX TNO(BCD)	READOUT ASEC	ASEC(BCD)	A_REC M
0	READOUT AFRAME	AFRAME(BCD)	A_REC L
0	0	SUB-Q CONTROL	SUB-Q CONTROL

_AH: TNO(BCD) (Valid when _AL is 0000_0010 or 0000_0011)

OUT AREG: Execution result
$00 OK
ELSE SUB ERROR CODE

Description Reads TOC information of the DISC currently attached. Maximum track number is 99.

[$0C] CD_CONTNTS: Initialize System Work

IN None

OUT AREG: $00 OK
ELSE SUB ERROR CODE

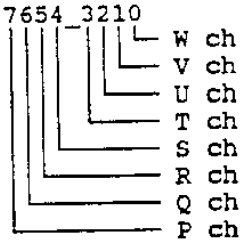
TNOMIN: Minimum track number (BCD)
TNOMAX: Maximum track number (BCD)
OUTMIN: Read out area start minute (BCD)
OUTSEC: Read out area start second (BCD)
OUTFRM: Read out area start frame (BCD)

Description The information listed above is normally set up during IPL. If DISC is changed during use, BIOS will not recognize this fact, so reset the system using this call. If this call is not used to notify BIOS of a change, BIOS calls will not be guaranteed.

[$0D] CD_SUBRD: Read All Channel Bits of Playing Subcode

IN None

OUT AREG: SUBCODE



CARRY: $00 SUBCODE is read
$01 SUBCODE not played

Description If drive is ready and disk is rotating, data contents are guaranteed.

[$0E] CD_PCMRD: Read Playing CD Linear PCM Data

IN AREG: $00 R-ch
$01 L-ch

OUT XREG: AUDIO DATA L
YREG: AUDIO DATA H

Description Read Linear PCM 16 bit data. One time read requires at least 25μsec.

[$0F] CD_FADE: Starting and Cancelling of Linear PCM and ADPCM Fade Out

IN AREG: Operation mode

$00	Cancel	FADE OUT
$08	PCM	FADE OUT (6.0 sec)
$0A	ADPCM	FADE OUT (6.0 sec)
$0C	PCM	FADE OUT (2.5 sec)
$0E	ADPCM	FADE OUT (2.5 sec)

OUT None

Description Once fade out is set, audio circuit will remain in a muting state until canceled. Cancel fade out before playing another song.

[$10] AD_RESET: Reset ADPCM Controller

IN None

OUT None

[$11] AD_TRANS: Transfer Data From CD to ADPCM Buffer

IN _CL: REC H
_CH: REC M
_DL: REC L

_AL: Number of transfer record

_DH: Transfer mode
$00 Set top address of destination
ELSE Transfer to current write point
(Address specification by _BX is unnecessary)

_BX: ADPCM buffer address

OUT AREG: Execution result
$00 OK
ELSE SUB ERROR CODE

[$12] AD_READ: Read Data from ADPCM Buffer to Memory

IN _CX: ADPCM Buffer address

_DH: Read address type

	00:LOCAL	FF:VRAM	2 ~ 6:MPR NO.
_BL:	ADR L	ADR L	BANK NO.($80 ~ $87)
_BH:	ADR H	ADR H	NO USE

OUT AREG: Execution result
$00 OK
ELSE ERROR

Description VDTIN_FLG changes to 1 during a read to VRAM. If MAWR or MARR is changed during this time because of an interrupt, etc., normal operation is not guaranteed.

[$15] AD_CPLAY: Continue ADPCM Playback

IN _CL: REC H
_CH: REC M
_DL: REC L

_AL: No. of playback record L
_AH: No. of playback record M
_BL: No. of playback record H

_DH: Sampling rate ($00 ~ $0E)
f KHz = 32 / (16 ~ _DH)

OUT AREG: Execution result
$00 OK
ELSE ERROR

Description Continues playback of ADPCM by regularly transferring 16 records of ADPCM data to the ADPCM buffer from the specified record. Minimum playback record number is 32 records. If read data, etc. is executed during repeated playback, head will move to a location more than 200 records away from the next ADPCM to be read or read operation will be retried. Then the ADPCM data cannot be read and playback will stop.

[$16] AD_STOP: Stop Data Playback ADPCM Buffer

IN None

OUT None

Description Stops ADPCM playback. Since playback speed is about 16KHz, a delay will occur before stopping. Check stop status through AD_STAT.

[$17] AD_STAT: Read ADPCM Controller Status

IN None

OUT AREG: ADPCM controller status
$00 ADPCM not busy (End or Not play)
ELSE ADPCM busy

XREG: ADPCM buffer and playback status
$00 Playing
More than a half buffer of data left
$01 Stop playback
$04 Playing
Less than a half buffer of data left

[$13] AD_WRITE: Write Data from Memory to ADPCM Buffer

IN _CX: ADPCM buffer address

_DH: Write data address type

	00:LOCAL	FF:VRAM	2 ~ 6:MPR NO.
_BL:	ADR L	ADR L	BANK NO. ($80 ~ $87)
_BH:	ADR H	ADR H	NO USE

_AX: Number of write size byte

OUT AREG: Execution result
$00 OK
ELSE ERROR

Description VDTIN_FLG changes to 1 during a read to VRAM. If MAWR or MARR is changed during this time because of an interrupt, etc., normal operation is not guaranteed.

[$14] AD_PLAY: Partial Playback of ADPCM Buffer Data

IN _BX: ADPCM buffer playback starting address

_AX: Number of playback size byte

_DH: SAMPLING RATE ($00 ~ $0E)
f KHz = 32 / (16 ~ _DH)

_DL: Mode

```
bit 7654_3210
    ?XXX_XXX?
      |    └── COUNTER MODE
      |          0: Set ADR, LENGTH, and RATE
      |          1: Set previous ADR, LENGTH, and RATE
      └─────── PLAY MODE
                0: AUTO STOP
                1: REPEAT
```

OUT AREG: Execution result
$00 OK
ELSE ERROR

Description This call will results in an error during AD-PCM playback. To ensure execution, stop playback by issuing AD_STOP. Then check to make sure playback is stopped using AD_STAT. If the status is okay, it is safe to issue the AD_PLAY command.

Back Up Memory Format

2KB battery back up memory is provided for IFU-30. This memory is commonly used by multiple applications, and is managed in the following format:

RAM file top

0000-0003	"HUBM" Back up memory ID
0004.w	RAM file size
0006.w	Unused area address
0008~000f	System reserved area

FCB format

+00.w	USER ID
+02.10byte	File name

DIR format

+00.w	Data block length + 10H
+02.w	Data block and FCB check sum
+04.12byte	FCB data
+16...	Data block

$8000 ~ $dfff are used as windows for back up memory access.

[$18] BM_FORMAT: Initialize Back Up Memory

IN _AX: Password Address

Password = "!BM FORMAT!"

OUT AREG: Execution result
$00 OK
$01 Cannot format

Description Prepare the password buffer and set its top address to _AX. If the password does not match, do not format.

[$19] BM_FREE: Check Free Back Up Memory

IN None

OUT _CX: Unused capacity in bytes

AREG: Execution result
$00 OK
$FF Format error

[$1A] BM_READ: Read Data from Back Up Memory

IN _AX: FCB address
_BX: Data read address
_CX: No. of bytes to read
_DX: Offset within file from top

OUT _CX: No. of bytes actually read

AREG: Execution result
$00 OK
$01 Cannot find file
$02 Bad data (Check sum error)
$FF Format error

[$1B] BM_WRITE: Write Data to Back Up Memory

IN _AX: FCB address
_BX: Data memory address
_CX: No. of bytes to write
_DX: Offset within file from top

OUT AREG: Execution result
$00 OK
$01 Not enough back up memory
$FF Format error

[$1C] BM_DELETE: Delete Data from Back Up Memory

IN _AX: FCB address

OUT AREG: Execution result
$00 OK
$01 Cannot delete
$FF Format error

[$1D] BM_FILES: Search Back Up Memory File

IN _BX: FCB destination address
_AL: Number from top of file (top = 1)

OUT AREG: Execution result
$00 OK
$01 Cannot find file
_AL = Largest file number
$FF Format error

[$1E] EX_GETVER: Get BIOS Version No.

IN None

OUT XREG: Integer part of version number
YREG: Decimal part of version number

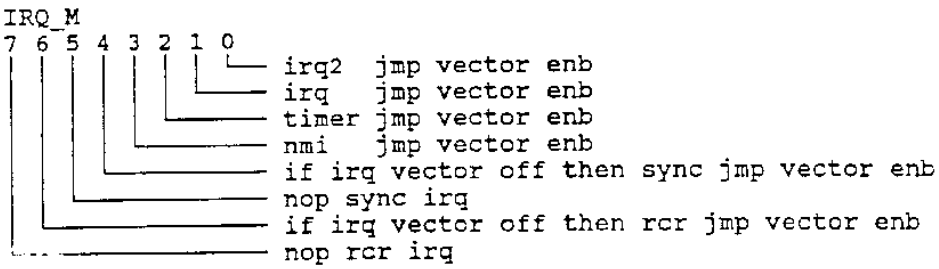
[$1F] EX_SETVEC: Set User Interrupt Vector Address

IN AREG: Vector number
$00 IRQ2
$01 IRQ
$02 TIMER
$03 NMI
$04 SYNC
$05 RCR
$06 SOFT RESET

XREG: Vector address low
YREG: Vector address high

OUT None

Description The above vectors are valid when its IRQ_M bit is ON. Turn on the specified bit after execution of this call. Operation is guaranteed if a bit is turned ON before setting the Vector.



Return from SYNC and RCR routines by using the rts instruction. Return from the others (IRQ2,IRQ, TIMER, and NMI) with the rti instruction.

To use the timer, the user must set the timer count and then turn the timer ON, after setting the vector bit on. The PSG driver in BIOS cannot be used during timer access, however, both can be used by calling PSG driver from the user timer routine. The PSG driver should be used with VSYNC access when used with the timer vector.

SOFT RESET VECTOR is the address branched out to when RUN+SELECT is pressed when IRQ in BIOS or the joy pad sense routine EX_JOYSNS is used.

[$20] EX_GETFNT: Transfer Kanji ROM Font Data (32 bytes)

IN _AX: Kanji code (SHIFT JIS CODE)

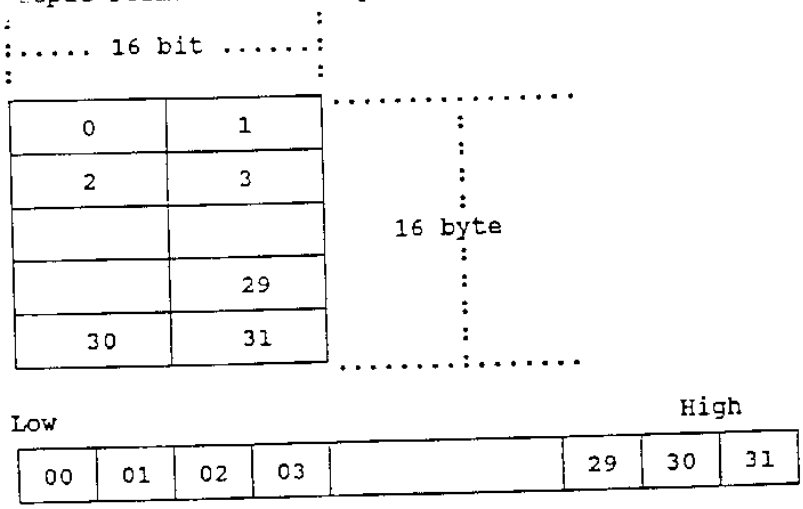
_BX: Transfer destination address

_DH: Transfer mode
$00 Transfer dot font 16x16
$01 Transfer dot font 12x12

OUT AREG: Execution result
$00 OK
$01 Kanji code error

Description This BIOS provides Kanji up to JIS level 1. $A000 - $DFFF or $6000 - $9FFF are used as the Kanji ROM window using the destination address.

Output Format for font pattern



[$21] EX_JOYSNS: Sense Joy Pad Status

IN JOYENA: Specifies software reset enable/disable.
Fixed to 1 when multi-tap is not used.
When multi-tap is used, refer to the following:

```
bit7654_3210
   XXX?_????  (Enable with bit ON)
      |||||
      ||||└── #1 pad
      |||└─── #2 pad
      ||└──── #3 pad
      |└───── #4 pad
      └────── #5 pad
```

OUT JOY,X: Senses ON/OFF
(X = 0 to 4)
JOYTRG,X: Sense trigger (OFF -> ON)
(X = 0 to 4)
JOYOLD,X: JOY contents of work one loop before
(X = 0 to 4)

```
bit7654_3210
   XXXX_XXXX
      ||||||||
      |||||||└── ONE    0:off 1:on
      ||||||└─── TWO    0:off 1:on
      |||||└──── SELECT 0:off 1:on
      ||||└───── RUN    0:off 1:on
      |||└────── UP     0:off 1:on
      ||└─────── RIGHT  0:off 1:on
      |└──────── DOWN   0:off 1:on
      └───────── LEFT   0:off 1:on
```

[$23] EX_SCRSIZ: Set No. of Characters in Virtual Screen

(Set to memory width register MWR)

IN AREG: Screen size (SCREEN;R09 bit4-6)

bit	2 1 0	x	y
	0 0 0	32	32
	0 0 1	64	32
	0 1 0	128	32
	0 1 1	128	32
	1 0 0	32	64
	1 0 1	64	64
	1 1 0	128	64
	1 1 1	128	64

OUT None

[$24] EX_ DOTMOD: Set VRAM Access Dot Width
(Set to memory width register MWR)

IN AREG: Dot size (VM;R09 bit0,1 SM;R09 bit 2,3)

OUT None

[$25] EX_SCRMOD: Set Screen Mode

IN AREG: Clock
$00 5MHz
$01 7MHz

XREG: Horizontal character size (do not use odd data)
10 ~ 34 (5MHz)
16 ~ 44 (7MHz)

YREG: Vertical character size
10 ~ 30 (5MHz, 7MHz)

OUT CARRY: 0 OK
1 ERROR

Description Default value is 5MHz, 32 x 30.

[$26] EX_IMODE: Set Increment Width for Memory Access
(1W;R05 bit 11,12)

IN AREG: 0 +1
1 +32
2 +64
3 +128

OUT None

[$27] EX_VMODE: Set VRAM Address Increment Width to 32,64,128 According to the Screen with EX_SCRSIZ.

IN None

OUT None

Description Set screen mode using EX_SCRMOD before using EX_VMODE. Use IRQ routine in BIOS.

[$28] EX_HMODE: Set VRAM Address Increment Width by 1

IN None

OUT None

Description Set screen mode using EX_SCRMOD before using EX_HMODE. Use IRQ routine in BIOS.

[$29] EX_VSYNC: WAIT VSYNC

IN None

OUT None

BREAK AREG

Description Use IRQ routine in BIOS. EX_VSYNC will return without waiting for VSYNC when Bit1 of IRQ_M is ON (while IRQ). If VSYNC IRQ is OFF, it is set ON.

[$2A] EX_RCRON: RASTER HIT IRQ ON (IE;R05 bit2)

IN None

OUT None

BREAK AREG

Description Use IRQ routine in BIOS.

[$2B] EX_RCROFF: RASTER HIT IRQ OFF (IE;R05 bit2)

IN None

OUT None

BREAK AREG

Description Use IRQ routine in BIOS.

[$2C] EX_IRQON: VSYNC IRQ ON (IE;R05 bit3)

IN None

OUT None

BREAK AREG

Description Use IRQ routine in BIOS.

[$2D] EX_IRQOFF: VSYNC IRQ OFF (IE;R05 bit3)

IN None

OUT None

BREAK AREG

Description Use IRQ routine in BIOS.

[$2E] EX_BGON: Background Display ON (BB;R05 bit7)

IN None

OUT None

Description Operates at a VSYNC interrupt after call. Use IRQ routine in BIOS.

[$2F] EX_BGOFF: Background Display OFF (BB;R05 bit7)

IN None

OUT None

Description Operates at a VSYNC interrupt after call. Use IRQ routine in BIOS.

[$30] EX_SPRON: Sprite Display ON (SB;R05 bit6)

IN None

OUT None

Description Operates at a VSYNC interrupt after call. Use IRQ routine in BIOS.

[$31] EX_SPROFF: Sprite Display OFF (SB;R05 bit6)

IN None

OUT None

Description Operates at a VSYNC interrupt after call. Use IRQ routine in BIOS.

[$32] EX_DSPON: Background and Sprite Display ON (SB,BB)

IN None

OUT None

Description Operates at a VSYNC interrupt after call. Use IRQ routine in BIOS.

[$33] EX_DSPOFF: Background and Sprite Display OFF (SB,BB)

IN None

OUT None

Description Operates at a VSYNC interrupt after call. Use IRQ routine in BIOS.

[$34] EX_DMAMOD: Set Operation Mode for DMA Transfer between VRAM & SATB and VRAMs. (DCR;ROF)

IN AREG: DMA mode

OUT None

[$35] EX_SPRDMA: Set Sauce Address and Transfer Between VRAM and SATB (DVSSR;R13)

IN SAT_ADR: SAT BUFF ADR L
SAT_ADR+1: SAT BUFF ADR H

OUT None

Description Use IRQ routine in BIOS.

[$36] EX_SATCLR: Clear Sprite Attribute Table (SAT) in VRAM

IN SAT_ADR: SAT BUFF ADR L
SAT_ADR+1: SAT BUFF ADR H

OUT SPRPTR = 0

Description Set VRAM increment width to 1.

[$37] EX_SPRPUT: Set Sprite Attribute to Where Specified by SAT in VRAM

IN SPRPTR: Definition number
SPRYL, SPRYH: Y-coordinate
SPRXL, SPRXH: X-coordinate
SPRNL, SPRNH: Pattern number
SPRAL, SPRAH: Attribute, Color

SATADR: SAT BUFF ADR L
SAT_ADR+1 SAT BUFF ADR H

OUT SPRPTR + =1

Description Set VRAM increment width to 1.

[$38] EX_SETRCR: Set Raster Detection Register (RCR;R06) Value

IN AREG: RCR L
XREG: RCR H

OUT None

[$39] EX_SETRED: Set to VRAM Read Mode

IN AREG: VRAM READ START ADR L
XREG: VRAM READ START ADR H

OUT None

[$3A] EX_SETWRT: Set to VRAM Write Mode

IN AREG: VRAM WRITE START ADR L
XREG: VRAM WRITE START ADR H

OUT None

[$3B] EX_SETDMA: Set Source(SOUR;R10), Destination Address(DESR;R11), and Transfer Block Length(LENR;R12) for DMA Transfer between VRAMs.

IN _BX: SOURCE ADDRESS
_DX: DESTINATION ADDRESS
_CX: BLOCK LENGTH

OUT None

[$4C] EX_COLORCMD: Set and Read Color Palette

IN	COLOR_CMD:	Function $00 Do nothing $01 Read color palette $02 Set color palette
	BGC_PTR:	Background color data buffer address L
	BGC_PTR + 1:	Background color data buffer address H
	BGC_LEN:	No. of background color palette
	SPRC_PTR:	Sprite color data buffer address L
	SPRC_PTR + 1:	Sprite color data buffer address H
	SPRC_LEN:	No. of sprite color palette
OUT	COLOR_CMD:	$00

About color palette setting

If IRQ routine in BIOS is used , color palette cannot be set through the above EX_COLORCMD method but in the following method.

Set the following values in the BIOS work area:

BGC_PTR:	Background color data buffer address L
BGC_PTR + 1:	Background color data buffer address H
BGC_LEN:	No. of background color palette
SPRC_PTR:	Sprite color data buffer address L
SPRC_PTR + 1:	Sprite color data buffer address H
SPRC_LEN:	No. of sprite color palette
COLOR_CMD:	Set function code 0: Do nothing 1: Read color palette 2: Set color palette

Operates at a VCYNC interrupt after COLOR_CMD is set. COLOR_CMD will be cleared to 0 when the operation is completed.

[$3C] EX_BINBCD: Switch Binary Value to BCD Value

IN	AREG:	Binary data
OUT	AREG:	BCD data
	CARRY:	0 OK 1 ERROR

[$3D] EX_BCDBIN: Switch BCD Value to Binary Value

IN AREG: BCD data

OUT AREG: Binary data
CARRY: 0 OK
1 ERROR

[$3E] EX_RND: Get Random Number

IN None

OUT AREG: Random number

Description The following system values need to be asynchronously changed to get a random number using this call.

RANDSEED	It will be incremented by each VSYNC interrupt if using IRQ in BIOS.
RNDM	Must be changed by user.

[$3F] MA_MUL8U: 8bit Multiplication with Unsigned Bit

8bit * 8bit = 16bit

IN _AL: Multiplicand
_BL: Multiplier

OUT _CX: Result

[$40] MA_MUL8U: 8bit Multiplication with Signed Bit

8bit * 8bit = 16bit

IN _AL: Multiplicand
_BL: Multiplier

OUT _CX: Result

[$41] MA_MUL16U: 16bit Multiplication with Unsigned Bit

16bit * 16bit = 32bit

IN _AX: Multiplicand
_BX: Multiplier

OUT _CX(L):_DX(H)

[$42] MA_DIV16U: 16bit Division with Unsigned Bit

16bit / 16bit = 16bit

IN _AX: Dividend
_BX: Divisor

OUT _CX: Quotient
_DX: Remainder

[$43] MA_DIV16S: 16bit Division with Signed Bit

16bit / 16bit = 16bit

IN _AX: Dividend
_BX: Divisor

OUT _CX: Quotient
_DX: Remainder

[$44] MA_SQRT: Square Root

IN _AX: Cardinal number

OUT _CL: Root

[$45] MA_SIN: Get Sine

IN AREG: Degree (DEGREE: 0° ~ 90°)

OUT CARRY: 0 AREG = AREG/256 Value
1 RESULT = 256

[$46] MA_COS: Get Cosine

IN AREG: Degree (DEGREE: 0° ~ 90°)

OUT CARRY: 0 AREG = AREG/256 Value
1 RESULT = 256

[$47] MA_ATNI: Get ATN

IN AREG: Y/X*64

OUT AREG: Value between 0° ~ 45°

[$48] PSG_BIOS: PSG Driver

IN _DH: Function number ($00 - $14)

Description See PSG driver document.

[$49] GRP_BIOS: Graphic Driver

IN _DH: Function number ($00 - $0F)

_DH

$00:VI_GINIT Initialize graphic BIOS

IN _AX: Graphic starting address
_BX: Paint stack area
OUT None

$01:VI_CASHCLR Clear graphic pointer cash. This is usually not used if initialized by VI_GINIT, shown above, which clears cash automatically.

IN None
OUT None

$02:VI_STRTADR Get graphic starting address and stack pointer.

IN None
OUT _AX: Graphic starting address
_BX: Stack pointer

$03:VI_GETADRS Get actual address of the coordinate address

IN _AH: X-coordinate
_AL: Y-coordinate
OUT _AX: Actual address
_BH: Bit number of byte (Count from 7th bit)

$04:VI_CLS Clear a 4 screen area currently specified by BAT top address for graphics.

IN None
OUT None

$05:VI_PSET Dot at the specified coordinates

IN _AH: X-coordinate
_AL: Y-coordinate
_DL: Color code (0 ~ 15)

OUT None

$06:VI_POINT Get color of a specified coordinate

IN _AH: X-coordinate
_AL: Y-coordinate

OUT AREG: Color code of the coordinate

$07:VI_LINE Draw a line between specified coordinates

IN _AH: X-coordinate of start point
_AL: Y-coordinate of start point
_BH: X-coordinate of end point
_BL: Y-coordinate of end point
_DL: Color code of line

OUT None

$08:VI_BOX Draw a box using the specified coordinates as a diagonal line

IN _AH: X-coordinate of start point
_AL: Y-coordinate of start point
_BH: X-coordinate of end point
_BL: Y-coordinate of end point
_DL: Color code of line

OUT None

$09:VI_BOXF Draw a box using the specified coordinates as a diagonal line and fill

IN _AH: X-coordinate of start point
_AL: Y-coordinate of start point
_BH: X-coordinate of end point
_BL: Y-coordinate of end point
_DL: Color code of line and fill

OUT None

$0A:VI_FLOOD Paint from the specified coordinate

IN _AH: X-coordinate of start point
_AL: Y-coordinate of start point
_DL: Color code of paint

OUT None

$0B:VI_PAINT Paint from the specified coordinate (different from the above FLOOD). Border line is specified and filled.

IN _AH: X-coordinate
_AL: Y-coordinate
_BH: Border line bit (H)
_BL: Border line bit (L)
_DL: Color code of paint

OUT None

$0C:VI_GWINDOW Set window at the specified coordinate

IN _AH: X-coordinate of start point (character unit)
_AL: Y-coordinate of start point (character unit)
_BH:
_BL:
_CH: No. of characters in X direction
_CL: No. of characters in Y direction
_DL: Color bank (not code)

OUT None

$0D:VI_GFONT Select font type for PUTFONT and SYMBOL output

IN _AL: Select type
0: 16 dot font
1: 12 dot font
2: 8 dot font
4: User defined character

OUT None

$0E:VI_PUTFONT Display Kanji at the specified coordinate

IN _AH: X-coordinate
    _AL: Y-coordinate
    _BX: Character font number
    _CL: Select back color and mode
        bit 0 ~ 3: Back color
        bit 6: Back transparent (1/0)
        bit 7: Fore transparent (1/0)
    _DL: Fore color

OUT None

$0F:VI_SYMBOL Display magnified specified character font

IN _AH: X-coordinate
    _AL: Y-coordinate
    _BX: Character font number
    _CL: Select back color and mode
        bit 0 ~ 3: Back color
        bit 4 ~ 5: Font (00:16dot, 01:12dot, 10:8dot, 11:user)
        bit 6: Back transparent (1/0)
        bit 7: Fore transparent (1/0)
    _CH: XY magnifying rate (x,y = 4/4bit, LOW/HI)
    _DL: Fore color

OUT None

[SUB ERROR CODE LIST]

$00 NO SENS
Does not have a valid sense byte. (Does not contain error information)

$04 NOT READY
CD-ROM drive not ready.

$0B NO DISC
CD not inserted.

$0D OPEN
Cover is open.

$11 DATA FIELD INCORRECT
More than one uncorrectable error occurred in one block.

$15 SEEK ERROR
Seek operation dod not complete within the specified time.

$16 HEADER READ ERROR
Cannot detect objective CD-ROM header.

$1C NOT DIGITAL AUDIO TRACK
Seek object address of the AUDIO TRACK SEARCH command is located within the data area.

$1D NOT CDROM DATA TRACK
Read address is not located within the CD-ROM data area.

$20 INVALID COMMAND
Received an unsupported command. (CDB top byte)

$21 INVALID ADDRESS
Invalid address or value within the CDB. (BCD check)

$22 INVALID PARAMETER
Invalid parameter within the CDB. (Other than an unused value or a specified value)

$25 END OF VOLUME
Specified address exceeds disk end location.

$2A INVALID PARAMETER LIST
Invalid value within the parameter list.

$2C NOT AUDIO PLAY STATE
Audio track not in play.

Note: CDB is an abbreviation of Command Data Byte.

```
;=============================================================
;
;    PC Engine CD-ROM SYSTEM BIOS  version 1.00
;
;    BIOS ENTRY and WORK equ
;
;    880913  Takaki Kobayashi
;
;=============================================================
;
max_mapping     equ       $FFF5

;-------------------------------------------------------------
;    BIOS ENTRY
;
cd_boot         equ       $E000
cd_reset        equ       $E003
cd_base         equ       $E006
cd_read         equ       $E009
cd_seek         equ       $E00C
cd_exec         equ       $E00F
cd_play         equ       $E012
cd_search       equ       $E015
cd_pause        equ       $E018
cd_stat         equ       $E01B
cd_subq         equ       $E01E
cd_dinfo        equ       $E021
cd_contnts      equ       $E024
cd_subrd        equ       $E027
cd_pcmrd        equ       $E02A
cd_fade         equ       $E02D

ad_reset        equ       $E030
ad_trans        equ       $E033
ad_read         equ       $E036
ad_write        equ       $E039
ad_play         equ       $E03C
ad_cplay        equ       $E03F
ad_stop         equ       $E042
ad_stat         equ       $E045

bm_format       equ       $E048
bm_free         equ       $E04B
bm_read         equ       $E04E
bm_write        equ       $E051
bm_delete       equ       $E054
bm_files        equ       $E057

ex_getver       equ       $E05A
ex_setvec       equ       $E05D
ex_getfnt       equ       $E060
ex_joysns       equ       $E063
ex_joyrep       equ       $E066
ex_scrsiz       equ       $E069
```

```
ex_dotmod       equ         $E06C
ex_scrmod       equ         $E06F
ex_imode        equ         $E072
ex_vmode        equ         $E075
ex_hmode        equ         $E078
ex_vsync        equ         $E07B
ex_rcron        equ         $E07E
ex_rcroff       equ         $E081
ex_irqon        equ         $E084
ex_irqoff       equ         $E087
ex_bgon         equ         $E08A
ex_bgoff        equ         $E08D
ex_spron        equ         $E090
ex_sproff       equ         $E093
ex_dspon        equ         $E096
ex_dspoff       equ         $E099
ex_dmamod       equ         $E09C
ex_sprdma       equ         $E09F
ex_satclr equ         $E0A2
ex_sprput       equ         $E0A5
ex_setrcr       equ         $E0A8
ex_setred       equ         $E0AB
ex_setwrt       equ         $E0AE
ex_setdma       equ         $E0B1
ex_colorcmd     equ         $E0E4
ex_binbcd       equ         $E0B4
ex_bcdbin       equ         $E0B7
ex_rnd          equ         $E0BA

ma_mul8u        equ         $E0BD
ma_mul8s        equ         $E0C0
ma_mul16u       equ         $E0C3
ma_div16s       equ         $E0C6
ma_div16u       equ         $E0C9
ma_sqrt         equ         $E0CC
ma_sin          equ         $E0CF
ma_cos          equ         $E0D2
ma_atni         equ         $E0D5

psg_bios        equ         $E0D8
grp_bios        equ         $E0DB

psg_drive       equ         $E0E1

;-----------------------------------------------------------------
;     ZERO PAGE WORK
;
zpgtop          equ         $20DC
;
zpg_grp_top     equ         $20DC
vi_bitpat       equ         $20DC           ;1
vi_rvbitpat     equ         $20DD           ;1
vi_ft_front     equ         $20DE
vi_padrs        equ         $20DE           ;2
```

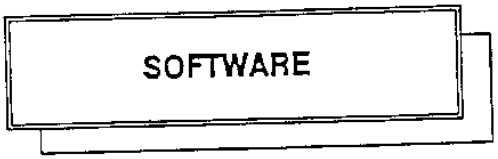
```
vi_porg         equ     $20E0           ;1
vi_ft_back      equ     $20E1           ;3
vi_stack        equ     $20E4           ;2
;
zpg_psg_top     equ     $20E6
time_sw         equ     $20E6           ;1
main_sw         equ     $20E7           ;1
si              equ     $20E8
si_l            equ     $20E8           ;1
si_h            equ     $20E9           ;1
r0              equ     $20EA
r0_l            equ     $20EA           ;1
r0_h            equ     $20EB           ;1
;
zpg_sys_top     equ     $20EC
zx0             equ     $20EC
zl0             equ     $20EC           ;1
zh0             equ     $20ED           ;1
zx1             equ     $20EE
zl1             equ     $20EE           ;1
zh1             equ     $20EF           ;1
zx2             equ     $20F0
zl2             equ     $20F0           ;1
zh2             equ     $20F1           ;1
cdi_b           equ     $20F2           ;1
crl_m           equ     $20F3           ;1
crh_m           equ     $20F4           ;1
irq_m           equ     $20F5           ;1
str_b           equ     $20F6           ;1
reg_box         equ     $20F7           ;1
_ax             equ     $20F8
_al             equ     $20F8           ;1
_ah             equ     $20F9           ;1
_bx             equ     $20FA
_bl             equ     $20FA           ;1
_bh             equ     $20FB           ;1
_cx             equ     $20FC
_cl             equ     $20FC           ;1
_ch             equ     $20FD           ;1
_dx             equ     $20FE
_dl             equ     $20FE           ;1
_dh             equ     $20FF           ;1

;-----------------------------------------------------------------------
;     RAM WORK
;
ramtop          equ     $2200
;
usrvec          equ     $2200
irq2_jmp        equ     $2200           ;2
irq_jmp         equ     $2202           ;2
tim_jmp         equ     $2204           ;2
nmi_jmp         equ     $2206           ;2
sync_jmp        equ     $2208           ;2
```

```
rcr_jmp         equ     $220A           ;2
bgx1            equ     $220C           ;2
bgx2            equ     $220E           ;2
bgy1            equ     $2210           ;2
bgy2            equ     $2212           ;2
sat_adr         equ     $2214           ;2
sprptr          equ     $2216           ;1
spry1           equ     $2217           ;1
spryh           equ     $2218           ;1
sprx1           equ     $2219           ;1
sprxh           equ     $221A           ;1
sprn1           equ     $221B           ;1
sprnh           equ     $221C           ;1
spra1           equ     $221D           ;1
sprah           equ     $221E           ;1
color_cmd       equ     $221F           ;1
bgc_ptr         equ     $2220           ;2
bgc_len         equ     $2222           ;1
sprc_ptr        equ     $2023           ;2
sprc_len        equ     $2025           ;1
joykeyflg       equ     $2226           ;1
joyena          equ     $2227           ;1
joy             equ     $2228           ;5
joytrg          equ     $222D           ;5
joyold          equ     $2232           ;5
irq_cnt         equ     $2241           ;1
notrdyflg       equ     $2247           ;1
rndseed         equ     $2249           ;1
rnd1            equ     $2249           ;1
rndh            equ     $224A           ;1
rndm            equ     $224B           ;1
tnomin          equ     $226A           ;1
tnomax          equ     $226B           ;1
outmin          equ     $226C           ;1
outsec          equ     $226D           ;1
outfrm          equ     $226E           ;1
vdtin_flg       equ     $2272           ;1
recbase0_h      equ     $2274           ;1
recbase0_m      equ     $2275           ;1
recbase0_l      equ     $2776           ;1
recbase1_h      equ     $2277           ;1
recbase1_m      equ     $2278           ;1
recbase1_l      equ     $2279           ;1
scsists         equ     $227B           ;1
suberrc         equ     $227C           ;1
sibcode         equ     $227E           ;1

ramend          equ     $22D0

psg_work_top    equ     $22D0
graph_work_top  equ     $2616
key_work_top    equ     $2649
user_work_top   equ     $267F
;---------------------------------------------------------------
```

```
;    PSG_BIOS FUNCTION NUMBER
;
PSG_ON          equ         0
PSG_OFF         equ         1
PSG_INIT        equ         2
PSG_BANK        equ         3
PSG_TRACK       equ         4
PSG_WAVE        equ         5
PSG_ENV         equ         6
PSG_FM          equ         7
PSG_PE          equ         8
PSG_PC          equ         9
PSG_TEMPO       equ         10
PSG_PLAY        equ         11
PSG_MSTAT       equ         12
PSG_SSTAT       equ         13
PSG_MSTOP       equ         14
PSG_SSTOP       equ         15
PSG_ASTOP       equ         16
PSG_MVOFF       equ         17
PSG_CONT        equ         18
PSG_FDOUT       equ         19
PSG_DCNT        equ         20

;-----------------------------------------------------------------
;    GRP_BIOS FUNCTION NUMBER
;
VI_GINIT        equ         0
VI_CASHCLR      equ         1
VI_STRTADR      equ         2
VI_GETADRS      equ         3
VI_CLS          equ         4
VI_PSET         equ         5
VI_POINT        equ         6
VI_LINE         equ         7
VI_BOX          equ         8
VI_BOXF         equ         9
VI_FLOOD        equ         10
VI_PAINT        equ         11
VI_GWINDOW      equ         12
VI_GFONT        equ         13
VI_PUTFONT      equ         14
VI_SYMBOL       equ         15
;-----------------------------------------------------------------
```

Hu7 CD SYSTEM
SOFTWARE MANUAL

TABLE OF CONTENTS


SOFTWARE

Chapter 1 SOFTWARE

The Hu7 CD system contains a 2HD floppy disk providing the following software. The software operates under MS-DOS ver3.1.

Description of terms

CD Contents	The Hu7 CD Contents Hard Disk Unit and the data area on the unit that contains the CD data is called CD Contents.
MS-DOS Partition	A part of the hard disk is secured for MS-DOS on the Hu7 CD contents hard disk unit. This is called the MS-DOS partition.
1) SCSIDEV.SYS	Device driver for an additional hard disk (620M), 8mm MT drive, and MS-DOS drive.
2) HDMUSIC.EXE	Records, plays, edits, and saves data for linear PCM. By using a bus mouse, this transfers data between the 8mm MT and CD contents, and between CD contents and MS-DOS.
3) ADPCM.EXE	Records, plays, edits, and saves data for ADPCM. By using a bus mouse, this creates MS-DOS data.
4) HDWRITE.EXE	Saves MS-DOS files to CD contents. It also adds, fetches, deletes, and lists files. It can be retrieved by batch file or by "make command" since it is a command line type of command.
5) CV.EXE	Converts the MX/BX file created by AS.LK to the file format (binary) used for CD.
6) INFGET.EXE	Converts file information saved in CD contents to the EQU file, which can be referenced by a program, and outputs this to the standard output.
7) CDEMUL.EXE	Emulates a CD-ROM2 drive through parallel I/O connected to the Hu7. Possesses monitor functions for the received commands.
8) BACKUPEX.EXE	Records files contained in CD contents to 8mm MT. MT can be saved as a master.
9) VERIFYEX.EXE	Compares the contents of CD contents to those of the 8mm MT. To verify the accuracy of the master, this must be executed. Print out the results to list the master contents.
10) RESTOREX.EXE	Restores data from 8mm MT to CD contents. This is used to restore data recorded by BACKUPEX.EXE to CD contents.

11) LISTEX.EXE	Displays a listing of the contents of the 8mm MT.
12) MSBACKUP.EXE	Records MS-DOS files to 8mm MT. Possesses add, fetch, and display list functions. Used for routine program BACKUP.
13) SFTFMT.EXE	Initializes the MS-DOS drive that is used by the CD contents unit. Format has already been done so use of this is not necessary.
14) EJECT.EXE	Opens the door of the 8mm MT.
15) HDTOC.EXE	Outputs TOC information of the CD. Print the result as a part of master documents.
16) HD384FMT.EXE	Executes physical format in case of an error on a hard disk.

Production Sequence

Chapter 2 PRODUCTION SEQUENCE

The following are the sequences required for creating various kinds of data and programs for CD-ROM2. Let's assume that this CD system is available for a programmer and a person who records linear PCM. The production sequence of program, picture data, ADPCM data, and linear PCM data are as follows:

2.1 Creating Programs

1. Creating IPL

   Create IPL and write data by using HDWRITE. (See IPL)

2. Estimating the data capacity and securing the program and data area

   Assume that plan and data structures are set. Estimate the capacity of each data as maximum. Because CD Contents Unit can only manage the programs and data as if it were on tape, secure the maximum capacity for program and data areas and organize them by name. For example, if you use 1M byte for your program, prepare as much as 2M bytes. 2MB can be controlled, but it will be better to divide it into 32K or 64K bytes. Files will be saved to CD contents using HDWRITE.EXE. (See HDWRITE.EXE)

   Example. HDWRITE -U10000, 0 PRG PRG1.BIN

   The above means that 10000H (64KB) is secured in CD Contents under the group name PRG, and the file name PRG1.BIN. You can repeat this 32 times to get 2M bytes. Secure the data area in the same way.

3. Getting record number

   Now, all programs and data areas are set in CD contents. You need to know the registered record number to get access to those programs and data. INFGET is the command for that. Always use this command to create a table.

4. Editing programs

   Create programs with an editor.

5. Creating CD files

   After creation of programs, use the assembler, linker, and converter to convert them to CD file format (binary). Use each tool as follows:

AS (Relocatable binary file)	: Assemble
LK (MX/BX file)	: Link
CV (Binary file)	: Convert

6. Writing files in CD

Write the files into CD Contents. With step 1, the area was secured for use, so execute HDWRITE to write files to CD Contents.

2.2 Creating Character Data

1. Creating data

Use various kinds of programs to create character data.

2. Creating CD files

Since the data format differs from program to program, convert files to binary files for creating CD files.

3. Writing files in CD

Write binary files into CD Contents. Execute HDWRITE at the secured area.

2.3 Creating ADPCM Data

1. Executing live recording

First, decide what you want to record, then record it using a cassette tape recorder.

2. Recording as ADPCM data

Record and save it as ADPCM data by using ADPCM.

3. Editing

If necessary, use multiple ADPCM data for editing.

4. Creating CD files

The data created at ADPCM is binary formatted. No change is necessary.

5. Writing files in CD

Write binary files into CD Contents. Execute HDWRITE at the secured area.

2.4 Creating Linear PCM Data

1. Executing live recording

   First, decide what you want to record. Then record it using a 2 track 38 tape recorder. As the linear PCM data is heard by a user as CD audio, a good recording system should be used.

2. Recording it as linear PCM data

   Record and register data as linear PCM data by using HDMUSIC.

3. Backing up to 8mm MT

   As the file volume of linear PCM data is too large to be handled by 2HD floppy disks, use 8mm MT.

4. Delivering the linear PCM data to a programmer

   A programmer saves the received data to CD contents by using HDMUSIC or he/she adds them using the RESTOREX A (append) option.

2.5 Creating a Master

1. Creating a master tape

   Record CD contents to 8mm MT by using BACKUPEX. It will take (depending on the volume) up to 2 hours.

2. Examining a master tape

   Compare the 8mm MT with CD contents using VERIFYEX. The list of contents of the 8mm MT will become a file.

3. Creating TOC list documents

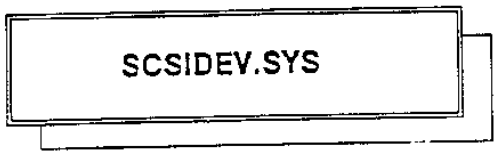
   Create the TOC list by using HDTOC.EXE and print out the document.

4. Executing a file listing

   Print out the file created by VERIFYEX. This is a part of the master document. Create three tapes as master tapes. One of them is for your records. Submit the other tapes as the masters. Write the necessary items in the BIOS usage information form and turn it in with forms 3.,4. and the two 8mm MT master tapes. To protect the master tapes, turn in a set of presented master verification forms as well.

   Master presentation documents and MT:

   1. File list
   2. TOC list
   3. 8mm MT - two tapes
   4. BIOS usage information form
   5. Presented master verification forms


SCSIDEV.SYS

Chapter 3. SCSIDEV.SYS

3.1 Outline

This file is a device driver operated by MS-DOS ver3.1, or higher, and it controls the 620MB hard disk and 8mm data cartridge ,etc. through SCSI interface.

3.2 System Requirements

Please note that this cannot work on machines manufactured before the PC-9801VX because of DMA.

3.3 Execution Sequence

This driver configures 6 software interrupts (INT command) from the number specified by the CONFIG.SYS file at start up time as an interface into MS-DOS. Therefore, it is necessary to specify the software interrupt at application.

3.4 Option Specification

A 298MB MS-DOS drive can be used as an option, but the maximum capacity for one drive is limited to 60MB by MS-DOS. Specification of software interrupt numbers is also possible.

3.5 Format

The MS-DOS partition operated by this drive cannot be initialized by the FORMAT program of MS-DOS. To construct a new system, to secure a new MS-DOS partition, or to change capacity, initialize with the attached SFTFMT command. (See SFTFMT)

3.6 Control Information

As this driver secures the 8MB MS-DOS partition as a control drive (used by system) at the end of 620M hard disk, the actual maximum capacity becomes 628MB.

3.7 Set Up CONFIG.SYS

To use this system, modify the CONFIG.SYS file into a device driver. The method of specification is the same as that for device driver specification. Specify path and file names of SCSIDEV.SYS. This device driver can specify the following options:

1. MS-DOS partition to be secured on 620MB hard disk

   There are two specification methods; to specify by mega byte, use a 1/2 sized number and a 1/2 sized letter [M] after / as in example 1. To specify by kilo byte, use a 1/2 sized number and a 1/2 sized letter [K] after / as in example 2. A maximum of 60M can be specified. The maximum drive number is 7, but the total volume capacity is 298M.

   Note: K is times 1024, and M is 1024 times of K.

   Example 1. DEVICE=A:¥SCSIDEV.SYS /40M
   Example 2. DEVICE=A:¥SCSIDEV.SYS /40960K

2. Specifying software interrupt numbers

   Specify them after the 1/2 sized [#] symbol by using 2 digit hexadecimal numbers. If this option is omitted, #68 will be specified. When using Japanese FP, specify software interrupt numbers not used by them.

3.8 Title at Execution

When this device driver is set up, the following titles will be displayed to tell you a device driver number, the secured drive, and its volume.

Example 1. When the following instruction is specified without two 2HD hard disks
DEVICE = A:¥SCSIDEV.SYS

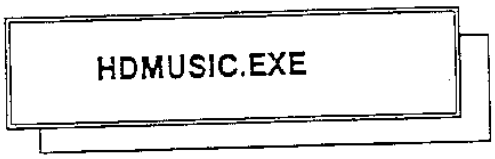
Translation: Can be used as a hard disk device driver Ver 1.00 (INT68H-6DH used) drive (C:8M)

```
ハードディスク　デバイスドライバー Ver 1.00 （INT 68H-6DHを使用）
ドライブ（C:8M ）として使用可能です
```

Example 2. When the following instruction is specified without two 2HD hard disks
DEVICE = A:¥SCSIDEV.SYS #50 /60M /20M /10M

Translation: Can be used as a hard disk device driver Ver 1.00 (INT50H-55H used) drive (C:8M D:60M E:20M F:10M)

```
ハードディスク　デバイスドライバー Ver 1.00 （INT 50H-55Hを使用）
ドライブ（C:8M D:60M E:20M F:10M ）として使用可能です
```


HDMUSIC.EXE

Chapter 4. HDMUSIC.EXE

4.1 Outline

This program has two functions. One is to record, playback, edit, and save linear PCM and the other is to back up and transfer files.

4.2 Linear PCM Data Processing Concept

The following five functions are provided for linear PCM and are done in CD contents:

1. Record

This is a function used to record onto a hard disk from audio tapes or other medi[illegible] Note that a sound length less than 4 seconds is not allowed when record-in[illegible]inear PCM on CD, so the minimum file size (4 sec) is 690K.

Warning: Recorded data is not set under system control without saving. Always save your important data. Refer to "Save" for details.

2. Playback

This is a function to transform a hard disk to a super CD player. It is much faster and better in quality than any other CD player in the market.

3. Edit

Basically this is a function to edit the area controlled on CD contents. This can edit not only music, but also data, programs, etc.

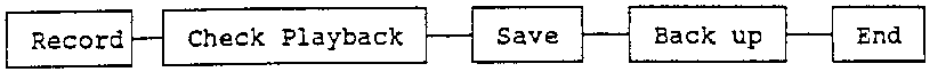
4. Save/Delete

Save/Delete is used to start or stop the handling of data (at the end of the hard disk) as CD data. In other words, "Save" puts data on the hard disk under system control and "delete" sets free the data from system control.

5. Visualize

This will visualize audio data using a wave form to graphically display the audio data.

Flow of linear PCM data creation



4.3 File Processing Concept

Normally, transfer from MS-DOS to CD Contents is done by using HDWRITE.EXE. However, HDWRITE.EXE is designed to receive all parameters from the command input of the shell, in case a big modification of the MAKE file takes place. So, for small changes or cases where visual control is desired, a function other than HDWRITE can be used. The following two functions are provided for file processes:

1. Related to 8mm MT

   Function to read and write data to 8mm MT. As a file volume is too big to handle linear PCM data for 2HD floppy, 8mm MT must be used.

2. Related to OS

   Function to playback, delete, rename, and list files of the CD Contents files.

4.4 Execution Method

Executes from the MS-DOS command line.

4.5 Option Specification

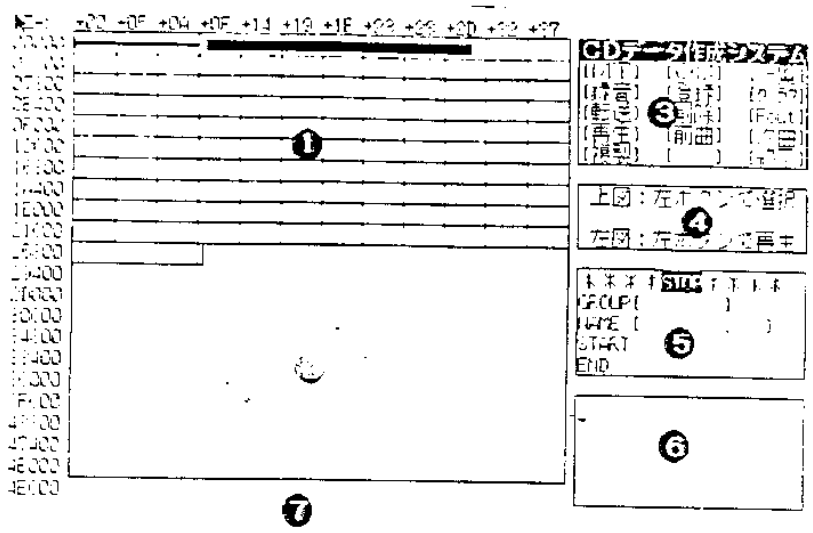
No option specification.

4.6 Format

HDMUSIC

4.7 Display at Execution

When HDMUSIC is executed, the screen looks like this:



4.7.1 Function of each area

1. CD contents area map:
   This is used to specify a record number. Left clicking roughly specifies a record number. Playback also starts from the location. Right clicking stops playing.

2. MS-DOS area:
   This is the area secured in the CD Contents Hard Disk Unit for the MS-DOS partition. Clicking here has no effect.

3. Command select area:
   Area to select each function. Clicking here executes each transaction.

4. Message line to users:
   A message from the program side is displayed here.

5. Status display and message line:
   Present CD player status such as playback, record, stop, or delete is displayed. When playback is executed, the record number of the starting record and the interrupted record number will be displayed.

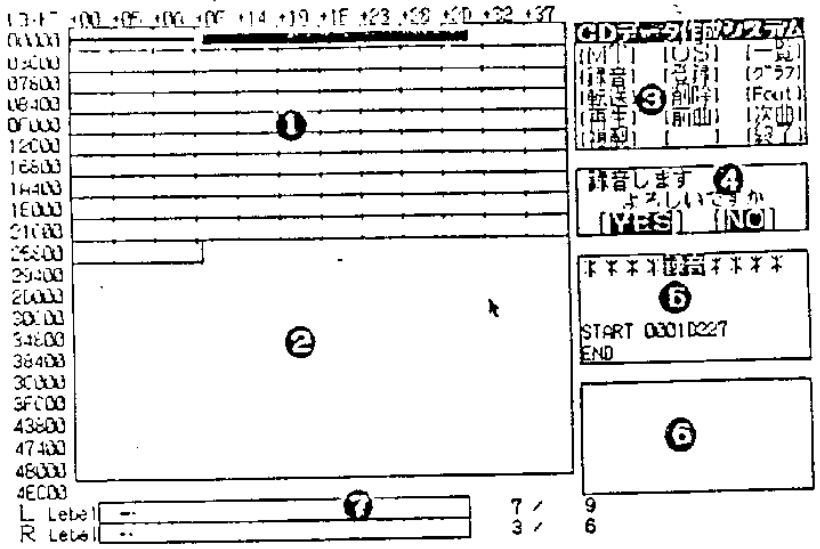
6. Message to be saved separately from area (5) such as register/transfer is displayed. Normally nothing is displayed here.

7. Plain graph:
   Normally nothing is displayed, but if necessary, a graph is displayed. This is to help you visually control the music.

4.8 Commands

4.8.1 Record

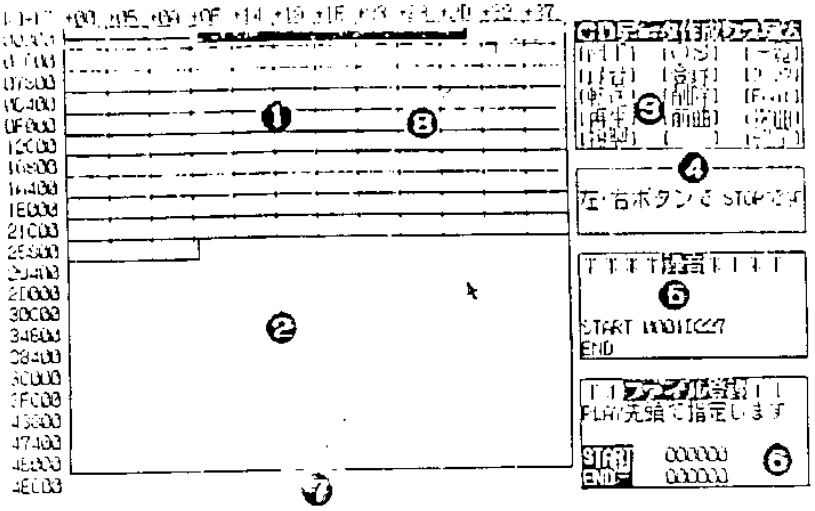
This is to record linear PCM data in the open area of CD Contents and to prepare it for saving.



1. Recording procedure

a) Click "record" in command select area (3) with the left button.

b) The above display appears and status is now "ready to record" showing the plain level meter shown at (7). Adjust levels by using the meter. The plain level meter indicates in db the value for left and right, separately. Color between cyan and red is -14db and scales are not set regularly. The number on the right side of the meter is digital absolute value of present value and maximum value. The number value would be 0 through 32768. Status display (5) becomes "record" and the starting record number to be recorded at START is displayed.



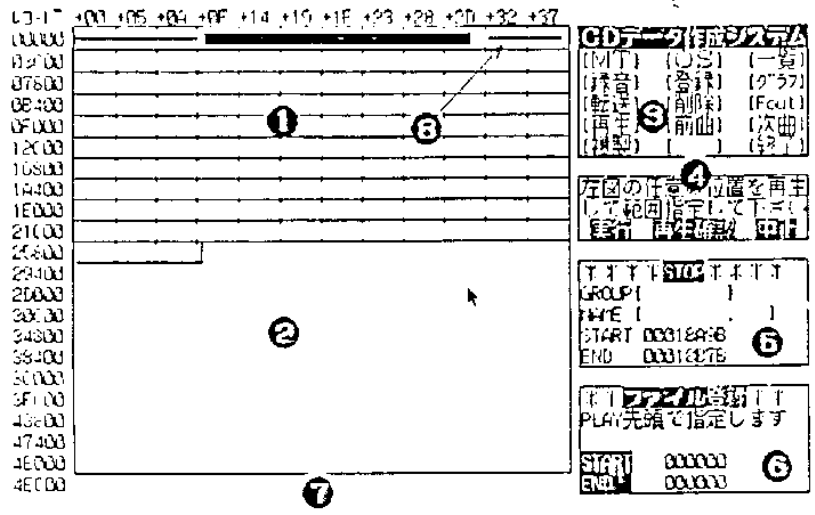
c) Start recording after level adjustment is completed. If YES is selected, the above display will appear and start recording. White lines such as (8) will increase in the area (1) while recording. If you need to cancel, select NO when the system asks a user to confirm Y/N at the message line, or a right click will stop this command. If retry is necessary, start again from a).

d) Either right or left click will stop recording. At this time, the white lined area will change into a red line. This shows the recorded area and lets you know about the next time for saving.

Note: If [MT] [OS] [List] [Graph] is selected, although these lines disappear, the contents will remain to allow you to save.

4.8.2. Save

This is to put PCM data on the hard disk (recorded or already registered) under system control and additionally save them as files. Recording is necessary before this command. (Always added to the end of the file control)



Saving Procedure

a) Click "register" in command select area (3) with the left button.

b) The above display appears with the area specified. The red reversed characters [Execute] [Check Playback] [Stop] will be displayed at the message line to the user (4).

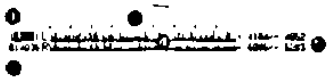
c) Specify the range you want to save. Specify starting and ending locations. There are two ways of specifying:

A. Click CD Contents (1) directly.

In this way, the record number clicked with the left button will be displayed in area (5). First, specify starting location to be saved in area (1) by clicking. Click "start" in area (6) with the left button to specify the start location. In the same manner, click the end location in area (1). Then, click "end" in area (6) with the left button to specify the end location.

B. Display plain graph (7).

First, click CD contents (1) directly to specify the appropriate location. Next, click plain graph (7) with the left button. The next graph will appear in "plain graph mode". You can specify detailed record numbers by looking at the graph indicator. The record number is displayed at the lower left of the plain graph. Refer to "plain graph" for details. Then, specify the starting location to be saved in plain graph. Next, click "start" in area (6) with the left button to specify the start location. In the same way, specify the end location in plain graph. Then, click "end" with the left button to specify the end location.



d) Confirm the area to be saved. Click "Check playback" with the left button to make sure that the area to be saved is correct. [illegible] the area to be saved will be played.

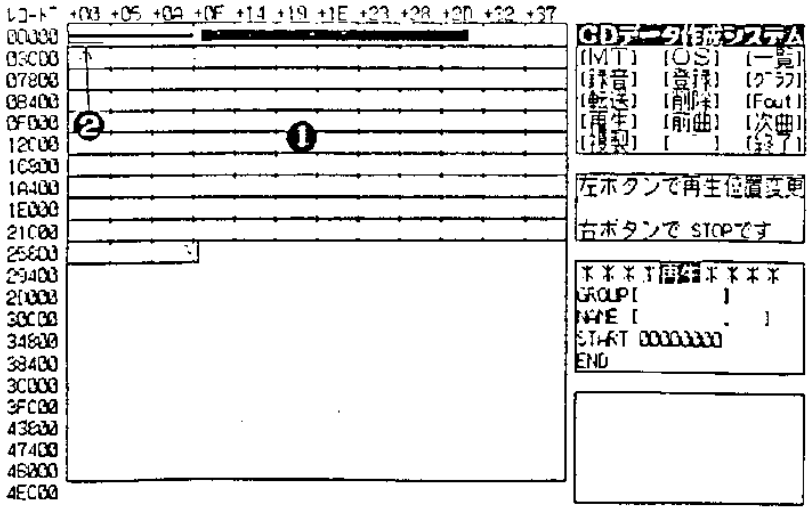
e) Saving. If the area that you want to save is correct, click "Execute" with the left button.

f) Save by using a group name or a file name. Type in group and name in area (6) for input. Files in CD Contents are stored under the group and file names. Specification of a group name is not necessary, but it is better to use the name for convenience.

g) Start saving. At this time, POST GAP (4seconds of blank) can be inserted right after the data file. It will take longer to save than to record.

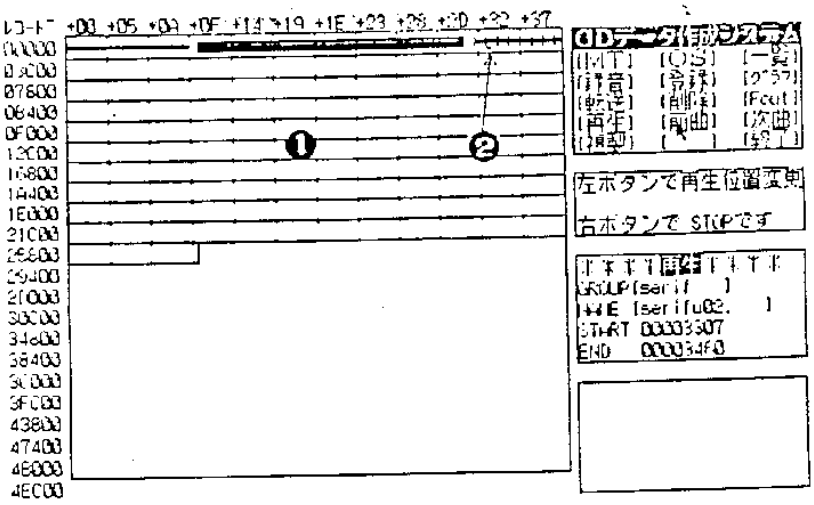
4.8.3 Playback

This is to playback the audio data from top, which is recorded in CD contents. The track is the one previously played. A left click of CD contents at (1) will change the playback location. This is used to listen to the recorded data. The current playback location is shown as white lines (2). To stop playback, click the right button.



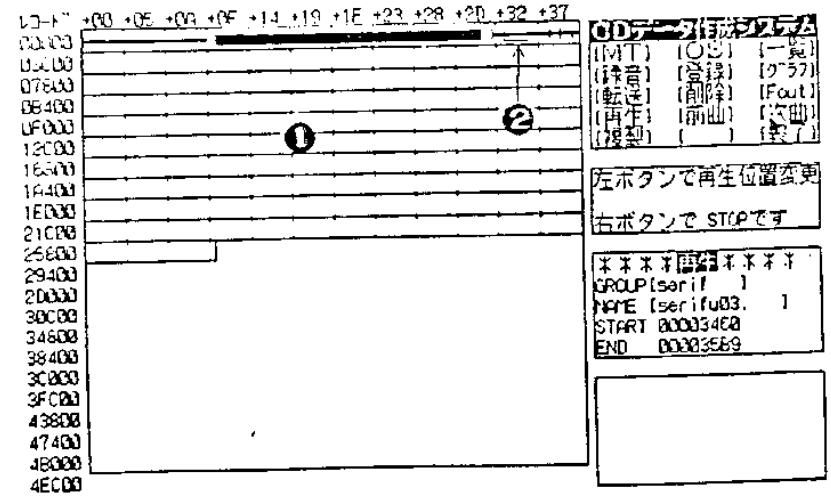
4.8.4 Previous song

This is to start playback from the top of the preceding song which played before the current song. Use this command to listen to the recorded contents. Current playback location is shown as white lines (2). To stop playback, click the right button.



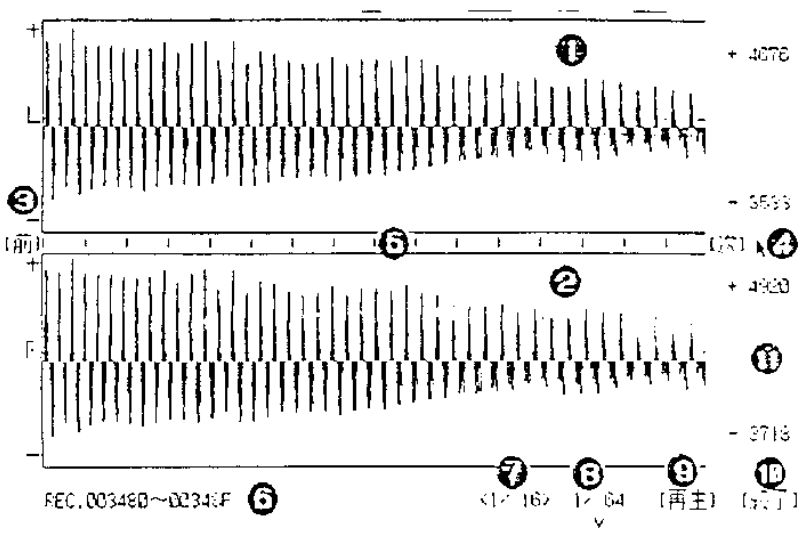
4.8.5 Next song

This is to playback from the top of the song next to track played before. Use this command to listen to the recorded contents. Current playback location is shown as white lines (2). To stop playback, click the right button.



4.8.6 Graph

This is to visualize PCM data on the hard disk. It starts graphing from the top of the preceding play. The display becomes as shown below:



Explanation of each area:

1. Graphic display area for the left channel
2. Graphic display area for the right channel
3. (Preceding) is the preceding block display
4. (Next) is the next block display
5. 2048 bytes unit scale. If area 1, 2, or 5 is clicked, start redisplay from the record.
6. Graphic record range
7. Sampling interval for graph data. 1/1 is continuous data, 1/16 is 16 pcs. (32 words) interval.
8. Scale of graph data. 1/1 is graph as it is, 1/64 is to reduce the graph to 1/64 size.
9. Clicking the left button plays the graphic range
10. Clicking the right or the left button terminates this command
11. Minimum and maximum value in the graphic record range

4.8.7 Transfer

This is to transfer data on the hard disk to another area. It can also transfer re-recorded linear PCM data to an area where data is already saved.

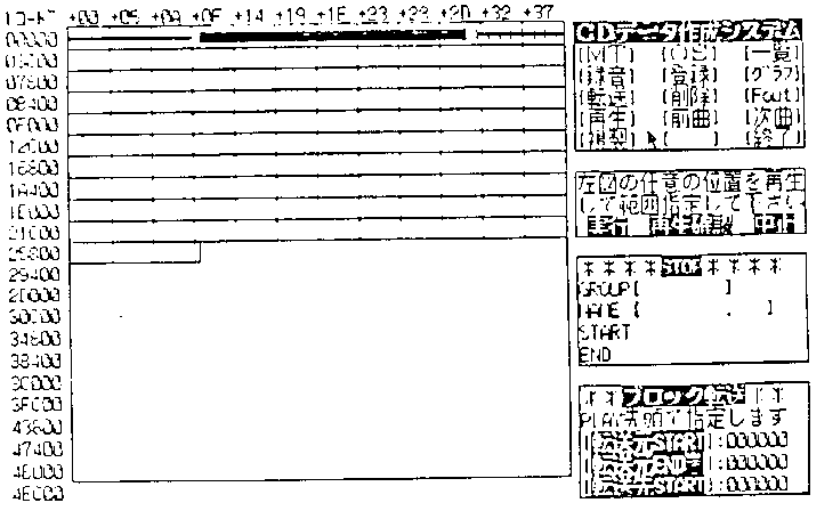
1. Operation procedure

a. The the same as for saving, decide a source START, END, and a destination START.

b. Check the source contents of the transfer by clicking "Playback" with the left button.

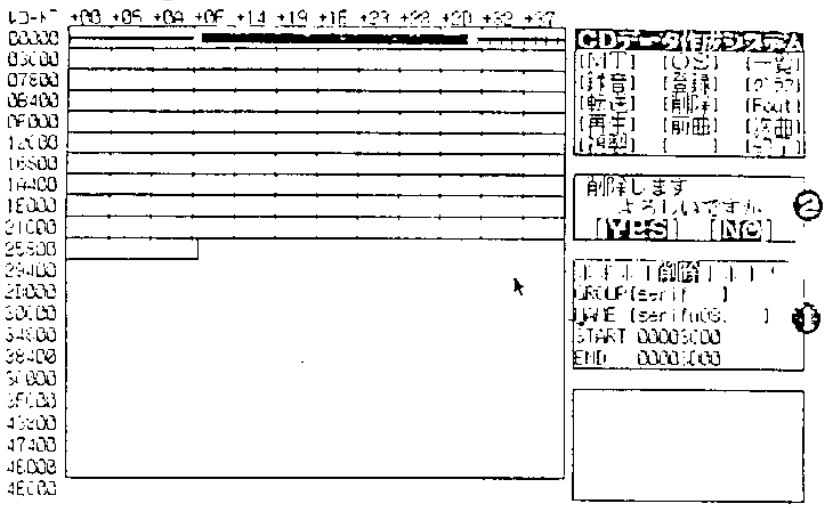
c. Click "Execute" with the left button to transfer.

If "Cancel" is selected, it will terminate automatically.



4.8.8 Delete

This is to completely delete the last file on a hard disk. It can only delete the end file. The display will be as follows and area (1) indicates group, name, start record, and end record, and area (2) confirms your command. Select "Yes" to delete, "No" to cancel. If "Yes" is selected, the file will be deleted and this command will end.



4.8.9 Fout

This is to fade out the linear PCM data on a hard disk. It reduces the play by 1 second less than what was specified. The last one second become 0 data, therefore, if play is less than 1 second, this does not function.

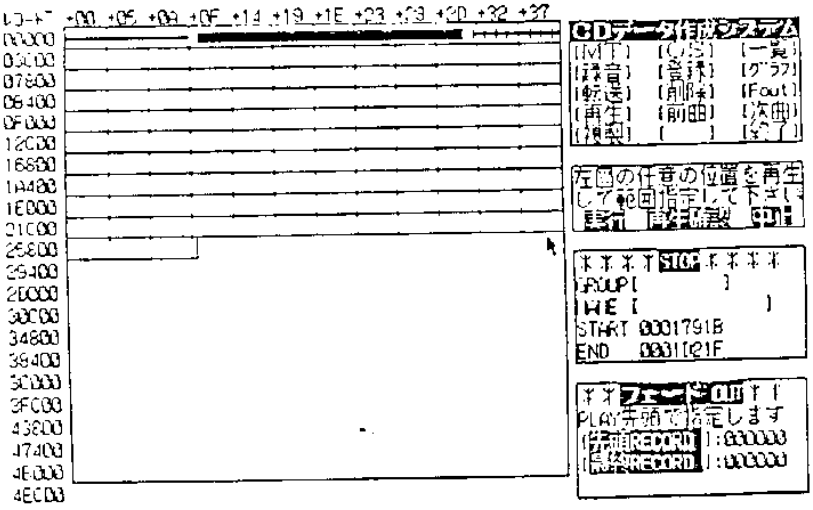
1. Operation procedure

a. The same as for saving, decide the starting record and the ending record by play top location specification.

b. Click "Play Check" with the left button and check the contents to fade out.

c. Click "Execute" with left button to fade out.

If "Ca[illegible]" is selected, it will terminate automatically.

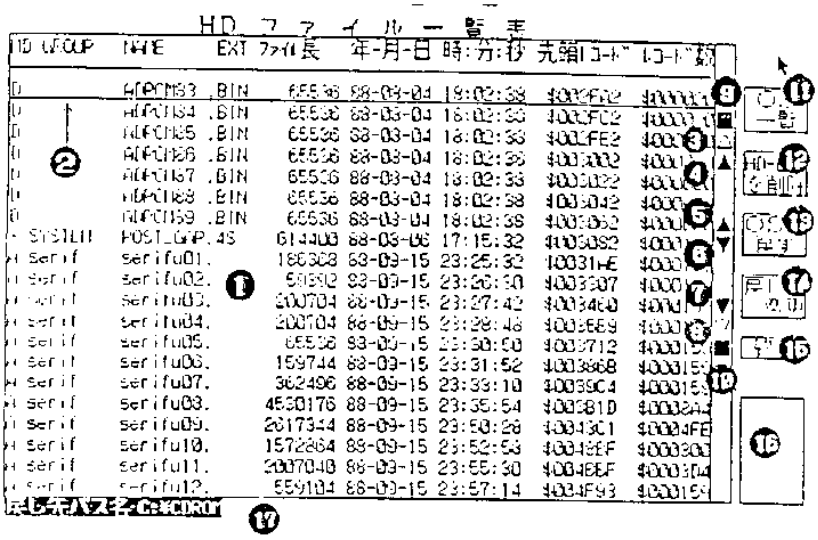


4.8.10 End

This is to end HDMUSIC. If "End" is clicked with the left button, HDMUSIC will end and return to the MS-DOS shell.

4.8.11 OS

This is to transfer files between MS-DOS and a hard disk. The following display will appear.



This is a list of files saved to CD contents. The files can be returned to the MS-DOS file one by one by specifying a MS-DOS path name. Each area has the following functions:

1. List of the file. Can be selected by a direct click of this area.
2. It indicates the files that are presently selected.
3. Displays two screens before
4. Displays one screen before.
5. Select one file before. If top, displays a listing of preceding files.
6. Select next file. If end, displays a listing of the next files.
7. Displays one screen after.
8. Displays two screens after.
9. Displays the first file.
10. Displays the end file.
11. Switches OS list and HD list.
12. Delete end file saved on HD.
13. Transfer currently selected file as a MS-DOS file to MS-DOS.
14. Change directory of destination.
15. Ends OS command and returns to the initial screen state.
16. Area where messages such as "Check" are displayed.
17. Displays the return destination path name.

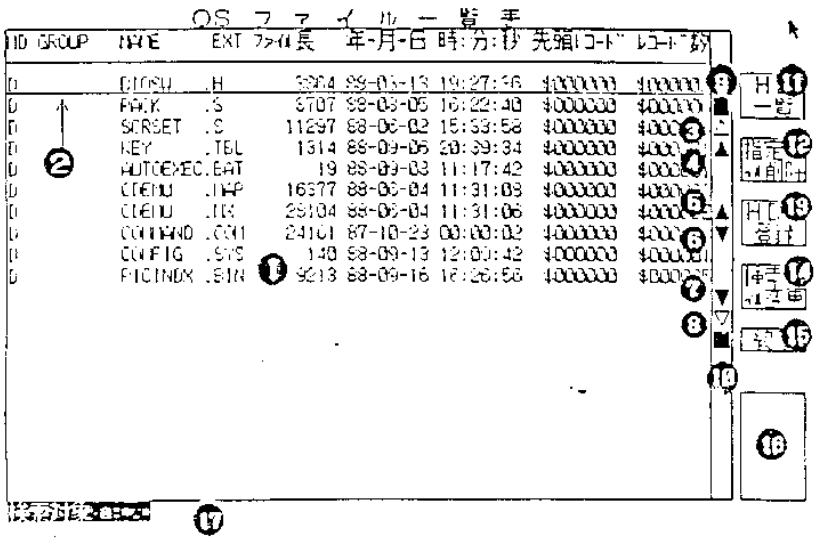
Application Method of This Command

1. Click (14) with the left button and specify a path name you want to return to.
2. Use (2) to (10) to specify files you want to return to MS-DOS.
3. Click (13) with the left button to return to MS-DOS.

About HD file item list

MD	Three file modes: D: Data A: Audio *: Delete file
GROUP	Group name
NAME	File name
EXT	Extension
File length	Number of valid bytes of a file
Y-M-S	Data when a file is saved
HH:MM:SS	Time when a file is saved
Starting record	Starting record number of a file on HD
No. of record	Number of records in files on HD

If (11) is clicked with the left button, the MS-DOS file list will be displayed, and files can be transferred between the MS-DOS file and CD contents. The following display will appear.



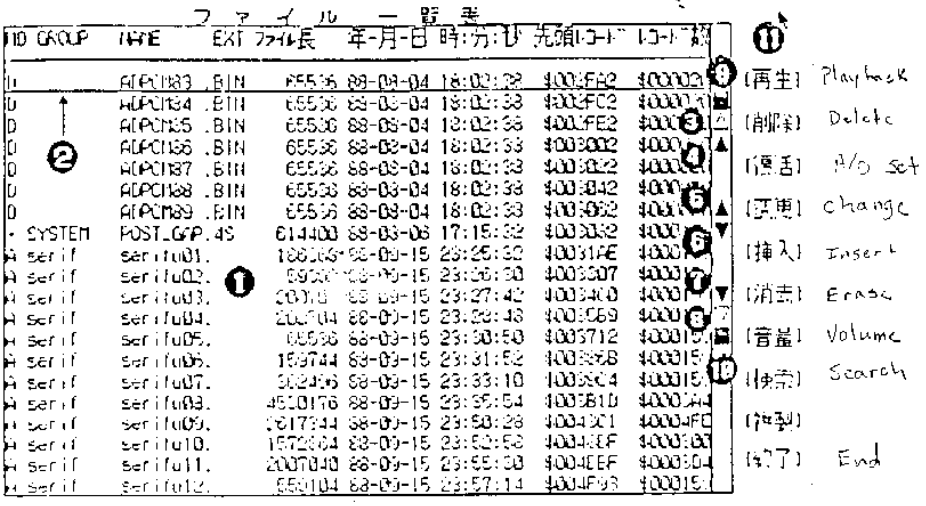
Specified MS-DOS files can be transferred to CD contents one by one.

Each area has the following function:

1. File list is displayed. Can be selected by direct clicking.
2. Indicates the currently selected file.
3. Displays a file list two screens before.
4. Displays a file list one screen before.
5. Select the previous file. If top, displays a listing of proceeding files.
6. Select the next file. If end, displays a listing of the next files.
7. Displays one screen after.
8. Displays two screens after.
9. Display the first file.
10. Display the end file.
11. Switches HD list and OS list.
12. Deletes the specified file.
13. Saves the currently selected file as a HD file.
14. Changes the selected files. MS-DOS wild cards can be used.
15. Ends OS command and returns to the initial screen state.
16. Message such as "Check" are displayed.
17. Displays the selected file.

4.8.12 List

This is to edit files saved in CD Contents. The following display will appear when this command is selected.



This is the file list saved in CD Contents. Each area has the following function:

1. File list is displayed. Can be selected by direct clicking.
2. Indicates the currently selected file.
3. Displays the file list two screens before.
4. Displays the file list one screen before.
5. Select the previous one file. If top, displays a listing of proceeding files.
6. Select the next one file. If end, displays a listing of the next files.
7. Displays one screen after.
8. Displays two screens after.
9. Display the first file.
10. Display the end file.
11. Command.

(Playback) Playback a file to select.

(Delete) Put a delete mark on the selected file. Actual data is not deleted.

(Change) Change the selected file name and group name.
After the following questions, type in the new information.
GROUP ?
NAME ?
EXT ?

(Insert) Insert the last file in front of the specified file.

(Erase) Erase the specified file and reorganize the files.

(Volume) Adjusts sound volume of the specified file. Use fraction (numerator 0 - 100, denominator 1 - 100 can be specified). If numerator is bigger than the denominator, it will take the biggest or the smallest value.

(Search) Displays a file list from the specified file. Inputs are group name, file name, and extension.

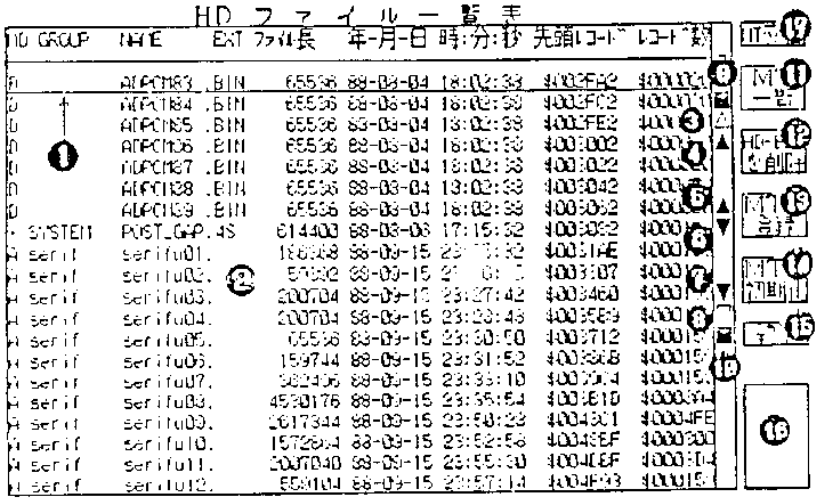
(End) Ends a file list and returns to the initial screen.

About File List Items

MD	Three file modes: D: Data A: Audio *: Delete file
GROUP	Group name
NAME	File name
EXT	Extension
File length	Number of valid bytes of a file
Y-M-S	Data when a file is saved
HH:MM:SS	Time when a file is saved
Starting record	Starting record number of a file on HD
No. of records	Number of records in files on HD

4.8.13 MT

This is to transfer a file between 8mm MT and a hard disk. Format for 8mm MT is the same as for the back up command, BACKUPEX.EXE. Operations such as changing the order of a 8mm MT back up of another system or returning a part of it to a hard disk are possible. The following display will appear:



This is the file list saved in CD Contents. Each specified file can be backed up to 8mm MT. Each area has the following function:

1. File list is displayed. Can be selected by direct clicking.
2. Indicates the currently selected file.
3. Displays the file list two screens before.
4. Displays the file list one screen before.
5. Select the previous one file. If top, displays a listing of proceeding files.
6. Select the next one file. If end, displays a listing of the next files.
7. Displays one screen after.
8. Displays two screens after.
9. Display the first file.
10. Display the end file.
11. Switches MT list and HD list.
12. Deletes the last file saved on HD.
13. Backs up the presently selected file to 8mm MT. If not initialized, this results in an error, so initialize MT in advance.
14. Initializes 8mm MT and brings it to the writable state. If 8mm MT is not set, this results in an error, so set it in advance.
15. Ends MT command and returns to the initial screen state.
16. Messages such as "Check" are displayed here.
17. Exchanges 8mm MT. Do not use the eject button on the Hu7 CD Contents Hard Disk Unit for exchanging 8mm MT. Always click MT exchange with the left button.

Operation

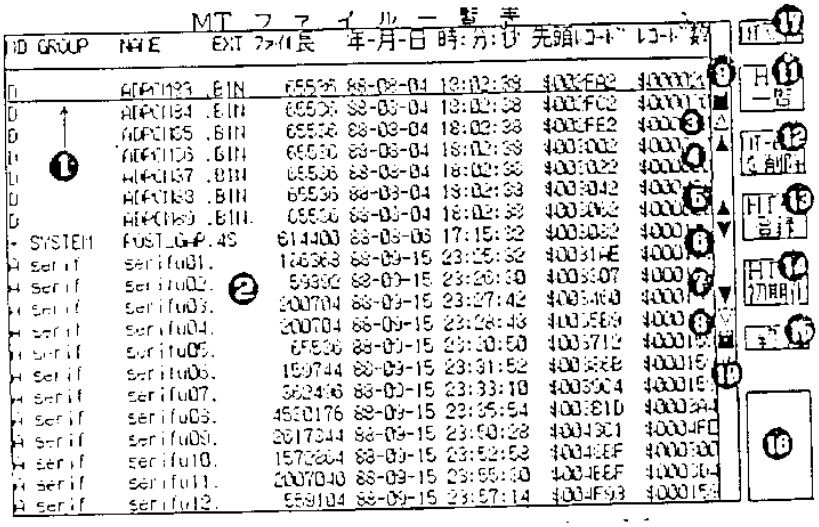
Application Method of This Command

1. Click (17) with the left button to open the 8mm MT door.
2. Set the 8mm MT and close the door.
3. Click (14) with the left button to initialize MT.
4. Specify a file to back up to 8mm MT by using (2) to (10).
5. Click (13) with the left button to back up to the 8mm MT.

About HD file list item

MD	Three file modes: D: Data A: Audio *: Delete file
GROUP	Group name
NAME	File name
EXT	Extension
File length	Number of valid bytes of a file
Y-M-S	Data when a file is saved
HH:MM:SS	Time when a file is saved
Starting record	Starting record number of a file on HD
No. of records	Number of records in files on HD

If the MT list (11) is clicked with the left button, the "Creating MT list" message will appear at (16) and starts reading 8mm. Once data starts coming into the 8mm MT, the following display will appear to list 8mm MT. Files can be transferred between 8mm MT and CD Contents.



If 8mm MT is not set, this results in an error. Always set it in advance.
Note: It takes a long time to create a 8mm MT list.

Each area has the following function:

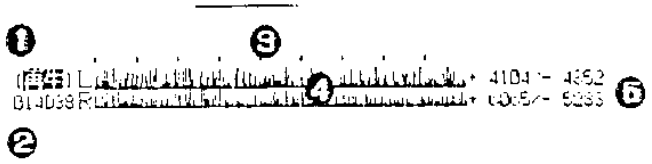
1. Displays a file list. Can be selected by direct clicking.
2. Indicates the currently selected file.
3. Displays the file list two screens before.
4. Displays the file list one screen before.
5. Selects the previous one file. If top, displays a listing of proceeding files.
6. Selects the next one file. If end, displays a listing of the next files.
7. Displays one screen after.
8. Displays two screens after.
9. Displays the first file.
10. Displays the end file.
11. Switches HD list and OS list.
12. Deletes the specified file.
13. Save the currently selected file as a HD file.
14. Initializes the Hu7 CD Contents Hard Disk (initialize control file only).
15. Ends the MT command and returns to the initial screen state.
16. Displays messages such as "Check" here.
17. Exchange 8mm MT. Do not use the eject button on the Hu7 CD Contents Hard Disk Unit to exchange 8mm MT. Always click MT exchange with the left button.

About MT file list items

MD	Three file modes: D: Data A: Audio *: Delete file
GROUP	Group name
NAME	File name
EXT	Extension
File length	Number of valid bytes of a file
Y-M-S	Data when a file is saved
HH:MM:SS	Time when a file is saved
Starting record	Starting record number of a file on HD
No. of records	Number of records in files on HD

4.6 Plain Graph

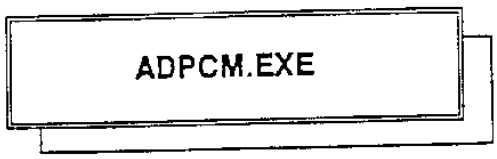
The plain graph mode is used to specify a record number in detail when you save or transfer files. A plain graph will appear if the plain graph display area (lower MS-DOS area) is clicked with the left button during playback. The display looks like this:



Each area has the following function:

1. Left click will start playback from record number at (2). During playback this plain graph disappears but it will reappear from the currently playing record number if (1) and (2) are clicked with the left button.
2. Displays the left end record number of the currently displaying plain graph. Left clicking this will also display 8 records before.
3. Indicates the record partition.
4. Plain graph. Specifies the record for start record if (3) and (4) are clicked with the left button.
5. Indicates the maximum value of the present graph.

If the CD Contents area map is clicked with the left button during plain graph mode, it will display a graph of the location.


ADPCM.EXE

Chapter 5 ADPCM.EXE

5.1 Outline

Playback of ADPCM is possible on the PC engine IFU30. This is to record, playback, edit, and save data for this ADPCM. These process are all done in the MS-DOS current directory. ADPCM is the method to compress a sound signal to 1 sample, 4 bits digital data.

Sampling frequency can be created with this program and data/second and the maximum time played at 64K are as follows:

Sampling frequency	Data/Sec.	Playback time
16 KHz	8000 bytes	8.192 seconds
8 KHz	4000 bytes	16.384 seconds
4 KHz	2000 bytes	32.768 seconds
2 KHz	1000 bytes	65.536 seconds

The following five functions are provided by this program:

1. Sampling frequency setting
   A sampling frequency of 16, 8, 4, 2 KHz can be set.

2. Record
   Record as ADPCM data. The recorded data is saved as a file. Cut the sound data from the file and save it as a regular file.

3. Playback
   Playback the data recorded as ADPCM from any data location.

4. Visualize (Graphic, HEX indication)
   It graphicizes ADCPM data for easier understanding. If ADPCM is well understood, HEX indication is possible.

5. Edit
   Lists ADPCM data and combines them to make a large file.

5.2 Execution Method

ADPCM.EXE is executed from the MS-DOS command line.

5.3 Format

ADPCM [Option specification] [Edit file names]

5.4 Option

/D Specifies a mode to edit in HEX data without graphics.
/N Specifies a mode without graphics nor HEX data. (Used only at recording)
-- Indicates how to use this program
-- The following display will appear if an option is specified.

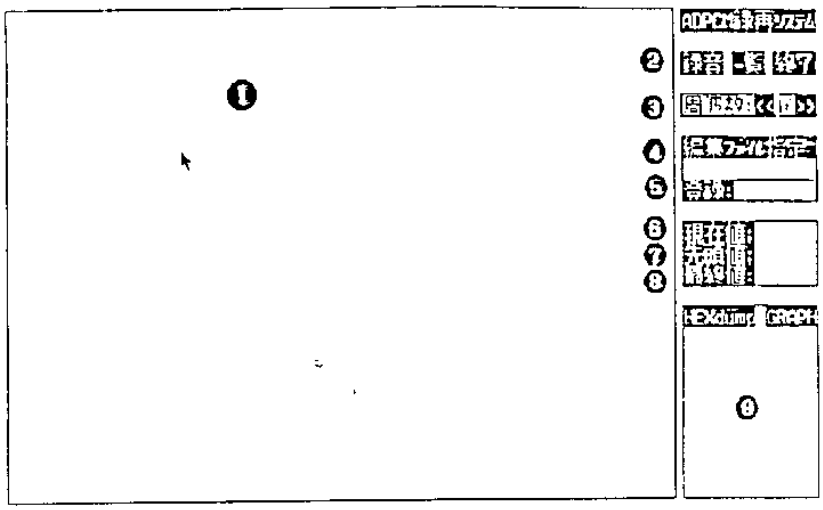
```
ADPCM Version 1.00     Copyright 1988 Hudson soft
使用法: ADPCM [/オプション] [編集ファイル名]
        /D      グラフ表示せずＨＥＸで編集します。
        /N      グラフ表示もＨＥＸ表示もしません。
        オプション省略はグラフ表示可能モードで、
        ファイル名省略は無指定となります。
        ＡＤＰＣＭの録音・再生・編集ツールです。
```

Translation:

Direction: ADPCM [/Option] [Edit file name]
/D Edit at HEX without graphics
/N Do not display graphics nor HEX
An omitted option results in graphic mode. An omitted file name results in no file name. This is a program for recording, playback, and editing.

The graphic mode (at omitted option) will convert ADPCM data to PCM data, and to drawing graphics from the data. The graphic mode enables editing data by looking at the graph. This mode takes time due to graphics. If ADPCM data mechanism is understood, the HEX mode can be used to cut time by specifying the [/D] option. The file name is the file name to be edited first. If specified, you can start editing immediately. If starting from recording, execute without specifying the file name.

The following display will appear if this program is executed without options.



Each area has the following functions:

1. Editable area. Graphic data will be displayed here. If clicked with the left button, ADPCM data will playback from the location.
2. Commands can be used by this program. Mode will change upon clicking the left button.
   (Record) Record ADPCM
   (List) Display a file list and execute playback and edit.
   (End) End the program and return to the MS-DOS command.
3. Sets sampling frequency. To decrease sampling frequency, click "< <" with the left button and to increase, click "> >" with the left button.
4. Specifies a file to edit. Clicking the left button brings a cursor to specify a file.
5. Saves. Execute after a start value (7) and an end value (8) are specified.
6. Indicates the playback start location as a present value.
7. Specifies a value to be used as start value at saving. If clicked with the left button, a present value (6) will be copied.
8. Specifies a value to be used as an end value at saving. If clicked with the left button, a present value (6) will be copied.
9. If the area (1) is too small, it can be magnified. If clicked with the left button, display the magnified record on (6). Clicking the right or left button at this area can make the starting location on the screen move back and forth.

5.5 Recording

Recording procedure is as follows:

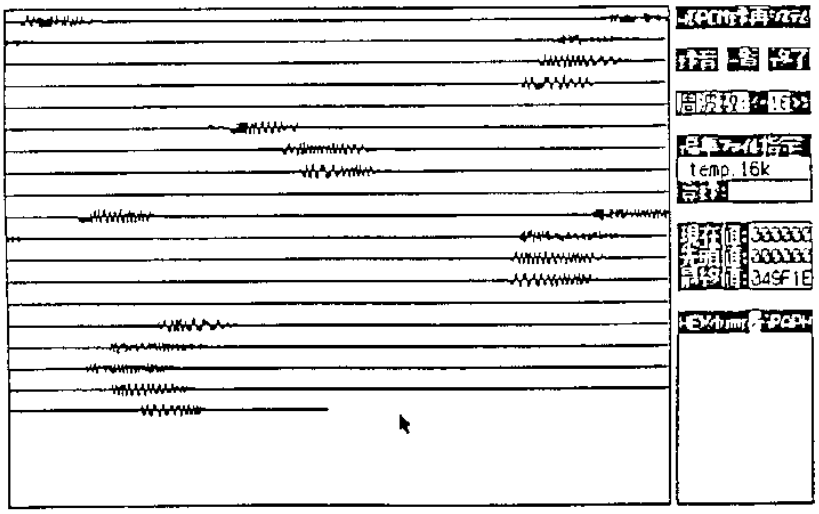
1. Live recording.
   First, decide what you want to record. Then record it using a cassette tape recorder, etc.

2. Set sampling frequency. Click "< <" or "> >" with the left button to set sampling frequency.

3. Clicking "Record" (2) with the left button will bring up the following indication at area (9), then prepare for recording.

   Record in TEMP.??K OK? [YES] /[NO]
   (?? is the sampling frequency at the time)

   If [Yes] is clicked with the left button, starts recording but areas with no sound will not be recorded. Clicking either left or right button will stop recording. If the level is exceeded during recording due to ADPCM exchange, "Level over" will appear in area (9). Clicking either button of the mouse will create graphs. It will take time for graphs to be displayed in area (1). If graphs are displayed, it will look like the following. If there is a red line, it means that the level is exceeded. If you play this bad data with the actual machine, the sound will be destroyed. Re-record or save the data except for areas with red lines.



5.6 Saving

You can separate and save a portion of data which has already been saved. The procedure is as follows:

1. Specify the location you want to save.

   Click the top of the wanted location with the left button to save. You may click with the left button as much as you want until you find exactly the right location. After the right position is found, click with the right button to interrupt playback and specify an end value by clicking with the left button.

2. Save files.

   If the right registration range is specified, save it as a file. Click "Register" (5) with the left button. Type the file name from the keyboard.

Using HEX dump and magnifying graphics.

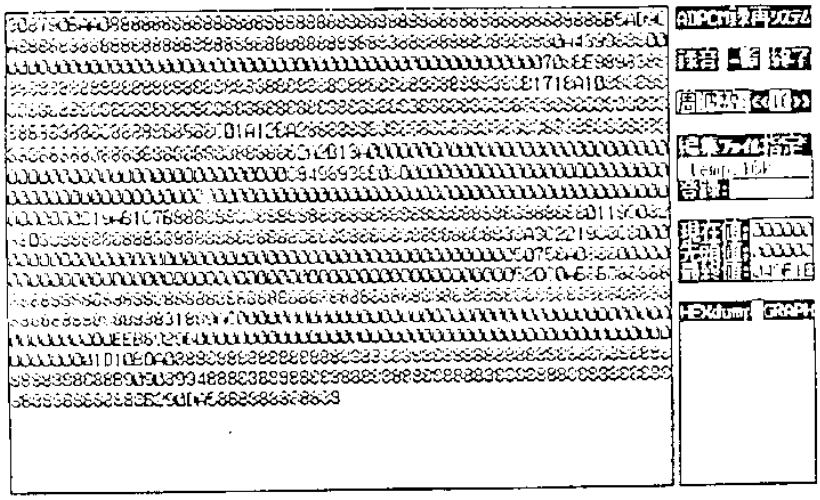
The area (9) can be magnified to see in detail if the right position cannot be specified in the editing area. Click (9) with the left button to bring up wave graphs. BACK is displayed at the left side of this area and NEXT at the right side of the area. Clicking the area with the left button will start displaying 64 samples before or after. Clicking wave form with the left button will display wave graphs. Click HEX dump with the left button from HEX dump/GRAPH will switch to the HEX mode. Mode can be switched at any time by left clicking. However, it cannot be used when the /D option is selected. HEX display will start by left clicking the HEX part.

3. Saving

   If the correct file name is input, it will be saved as a MS_DOS file. However, if an extension is not specified, an extension indicating sampling frequency such as 16K, 08K, 04K will be added automatically.

5.7 HEX Mode

Execute a HEX dump without wave graphics, if the /D option is specified. Use this when you understand the ADPCM data format. HEX data will be displayed instead of wave graphics after completion of recording. HEX data will be reduced according to the size of files since it cannot be displayed in the whole edit area. Others do not function specifically. The following screen appears when recording with the /D option.

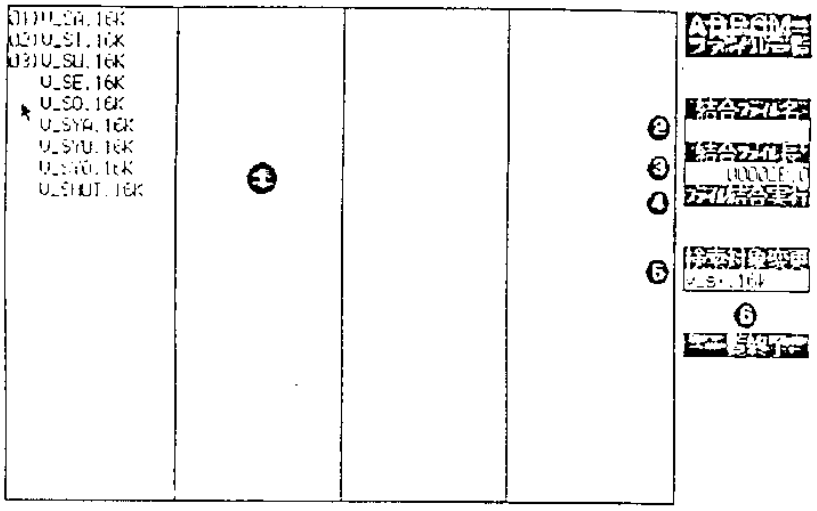


5.8 List

The list mode of ADPCM has the following three functions:

1. Display a list
2. Playback files by unit
3. Combine files
4. Get information of combined files

The following screen will appear if Display list is clicked with the left button.



Each area has the following functions:

1. Total of 91 files can be displayed in the area. (The excess will be cut) The format is three blank spaces, a file name, and extension. The following two functions can be done in this area:

   a. Playback

   Playback ADPCM data by reversing the file name by clicking the file name with the left button. (Sampling frequency will be read automatically by the extension (??K).) Pressing the mouse button will stop it during playback.

   b. Assign order of combination

   Order of combination can be assigned by clicking on the three letter area of the file name. Another click will erase the order. Erase from a larger number. If the order is assigned, the file length (3) will increase. If erased, the file length will decrease.

2. Input the file names to be combined. Clicking the left button will input file names. You can only specify file names here.

3. Display the file length of combined files in hexadecimal notation. Adjust not to go beyond 10000 bytes since the ADPCM buffer is 64KB. (This program will not warn you if it is over 64KB.)

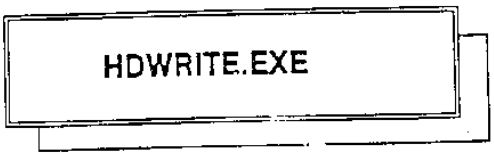
4. Start combining files by clicking "File combine execute" with the left button. If a file name is not specified, then a window will appear asking for the input file name. The combined files will be output to the specified file and the combined information file will be output to the name specified using .H extension. The contents are as follows:

   Example of combined information files

```
TEST1_BX EQU $0000     CONTAINED FROM BEGINNING
TEST1_AX EQU $1023     $1023 BYTES
TEST1_DH EQU $0E       FREQUENCY 16K

TEST2_BX EQU $1023     CONTAINED FROM ADDRESS $1023
TEST2_AX EQU $0800     $800 BYTES
TEST2_DH EQU $0C       FREQUENCY 8K
```

5. To specify file names to be listed, click "change selection" with the left button. Use wild card characters * and ? to specify.

6. To return to the main menu, click "list end" with the left button.


HDWRITE.EXE

Chapter 6. HDWRITE.EXE

6.1 Outline

This is to save additional MS-DOS files to CD Contents or to fetch them. Can be used by MAKE or by batch file since it is a command line type program that does not use a mouse. The following six functions are provided:

1. Save files to CD Contents
2. Update files on CD Contents
3. Fetch files from CD Contents
4. Delete specified file
5. Delete files after a specified file
6. Output a file list

6.2 Execution Method

HDWRITE.EXE is executed from the MS-DOS command line.

6.3 Format

HDWRITE [Option specification] Path name [Group name] [File name]

6.4 Option

-U [M,N]	Add and update files. [M] specifies a maximum value for file length. [N] specifies offset in 800H units. A value for both [M] and [N] are contained in 8 digits HEX.
-X	Fetch files
-D	Put a delete mark on files
-K	Delete files after a specified file
-L	Output a file list

The following display will appear if HDWRITE is executed without any parameters.

```
HDWRITE Version 1.00 Copyright 1987 Hudson soft
使用法：HDWRITE ［スイッチ］ パス名 ［グループ名］ ［ラベル名］
        -U[M,N] ファイルの追加・更新、[M]は最大値、[N]はオフセット
                16進 8桁のバイト値指定、[N]は800Hの倍数であること
        -K      指定ファイル以後を全て削除する
        -D      ファイルに削除マークを付ける
        -X      ファイルの取り出し
        -L      一覧出力
```

Translation:

Direction : HDWRITE [Switch] Path name [Group name] [Label name]

-U[M,N]	Add or update files. [M] is a maximum value for the file, [N] is the offset in hex notation specified by an 8 digit value. [N] is a multiple of 800H.
-K	Delete files after a specified file
-D	Put a delete mark on files
-X	Fetch files
-L	Output a file list

Each command area has the following meaning:

Path name: MS-DOS drive:Directory¥File name.Extension
Wild cards can be specified.

Group name: Consider this as a part of the file name managed by CD Contents. If this is not specified, it is treated as if there is no group specification.

Label name: File name.Extension managed in CD Contents. If this is not specified, the file name (File name. extension) will be used. If a wild card is specified at path name, do not specify a label name. A label name cannot be specified with the -X,-D, or -K options, because a path name equals a file name.extension for CD Contents.

6.5 Directions

We will give an example of actual use.

To save a new file to CD Contents

Name a MS-DOS file with group name [TEST] and save it under the name of [TEST.BIN] .

```
E>HDWRITE TEST.BIN TEST
HDWRITE Version 1.00 Copyright 1987 Hudson soft
WRITE:D TEST TEST .BIN 17137 88-07-11 16:47:28 $001000 $000009
```

If a file [TEST.BIN] of [TEST] has already been saved and its volume is equal to or smaller than the previous version, update the file. If the volume becomes larger, a delet mark will be placed at the old location and it will be added in at the end location. If it is new file, add it at the end location. To write a new file (especially program or character data, etc. which is likely to change), save the maximum volume using the U option so that there will be no need to change it later.

To fetch a file from CD Contents

Return a file in CD Contents called [TEST.BIN] and group name [TEST] to a MS-DOS file [TEST.BIN] under the current directory.

```
E>HDWRITE -X TEST.BIN TEST
HDWRITE Version 1.00 Copyright 1987 Hudson soft
READ:D  TEST TEST .BIN 17137 88-07-11 16:47:28 $001000 $000009
```

To delete the specified file

Assume that the CD Contents file [TEST.BIN] and the group name [TEST] have been deleted.

```
E>HDWRITE -D TEST.BIN TEST
HDWRITE Version 1.00 Copyright 1987 Hudson soft
DELETE:D TEST TEST.BIN 17137 88-07-11 16:47:28 $001000 $000009
```

To delete files after the specified file

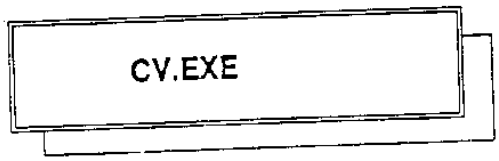
Delete all files after the CD Contents file (TEST.BIN), group name (TEST).

```
E>HDWRITE -K TEST.BIN TEST
HDWRITE Version 1.00 Copyright 1987 Hudson soft
KILL:D  TEST  TEST.BIN 17137 88-07-11 16:47:28 $001000 $000009
Delete following files.  OK(Y) ?Y
Deleted
```

Nothing will happen if a key other than Y is pressed.

To list files

```
E>HDWRITE -L
HDWRITE Version 1.00 Copyright 1987 Hudson soft
MD_GROUP__NAME___.EXT_BYTES__YY/MM/DD_HH:MM:SS_RECORD__LENGTH
A CD_ROM MESSAGE.     7913472 88-02-24 18:57:02 $000000 $000F18
* SYSTEM PRE_GAP.3S    475136 56-02-04 02:02:48 $000F18 $0000E8
D TEST   TEST    .BIN   17137 88-07-11 16:47:28 $001000 $000009
D TEST   HS1     .CG    29696 88-06-04 15:12:32 $001009 $00000F
D TEST   HS2     .CG    29696 88-06-04 15:12:34 $001018 $00000F
D TEST   AME0    .CG    29696 88-06-04 15:12:34 $001027 $00000F
MD_GROUP_NAME___.EXT___BYTES_YY/MM/DD_HH:MM:SS__RECORD__LENGTH
```


CV.EXE

Chapter 7 CV.EXE

7.1 Outline

This is to convert [.MX] /[.BX] files created by AS/LK to CDROM data image files. Use this converter to make binary files for program files and write the files to CD Contents by using HDWRITE or HDMUSIC.

7.2 Execution Method

CV.EXE is executed from the MS-DOS command line.

7.3 Format

CV Input file name Output file name

7.4 Options

No options available

7.5 Directions

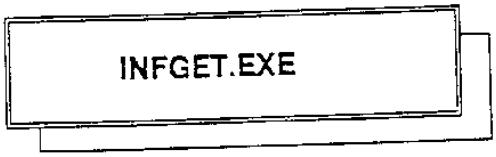
The following direction will be displayed if input and output file names are not specified before execution.

```
E>CV
binary file converter Ver 1.00 copyright 1988 Hudson scft
Usage: cv in_file out_file
```

The in_file is a [.MX] /[.BX] file name output by AS/LK, and out_file is a binary file name for CD ROM.

Example. To convert DSP.MX to DSP.BIN

```
E>CV DSP.MX DSP.BIN
binary file converter Ver 1.00 copyright 1988 Hudson soft
address = 00004000 length = 02e0
```


INFGET.EXE

Chapter 8. INFGET.EXE

8.1 Outline

This is to output file information saved in CD Contents as a EQU file that corresponds to the program. It can be redirected to create files to send to standard output.

8.2 Execution Method

INFGET.EXE is executed from the MS-DOS command line.

8.3 Format

INFGET [Option]

8.4 Options

-A Data record also repeats absolute record from a start
-E Extension is also a part of the label name
-G Group name is also a part of the label name
-- Displays directions

```
E>INFGET--
Directions : INFGET - switch character
      Write CD system control information to standard text
      -A  Data record repeats absolute record from start.
      -E  Extension is also a part of label name.
      -G  A group name is also a part of label name.
```

Output format is as follows:

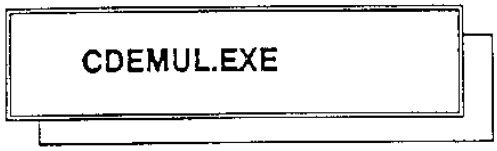
Label name _ SM EQU Start specification of music at label name (Minutes) BCD data
Label name _ SS EQU Start specification of music at label name (Seconds) BCD data
Label name _ SF EQU Start specification of music at label name (Frame) BCD data
Label name _ EM EQU End specification of music at label name (Minutes) BCD data
Label name _ ES EQU End specification of music at label name (Seconds) BCD data
Label name _ EF EQU End specification of music at label name (Frame) BCD data
Label name _ HI EQU Most significant byte of start record of the data at label name
Label name _ LW EQU Least significant 2 bytes of start record of the data at label name
Label name _ LN EQU Length of data is given in 2 bytes at label name *
Label name _ RL EQU Least significant 2 bytes of the number of records is given at label name *
Label name _ RH EQU Most significant byte of the number of records is given at label name *

* Output only more than 64K record.

8.5 Directions

This command is usually used with the output redirected to files.

```
A>INFGET >TEST.H

(TEST.H CONTENTS)

;TRACK NO.=01

;88-02-24          18:57:02                    CD_ROM_MESSAGE.
MESSAGE_SM  EQU    $00
MESSAGE_SS  EQU    $02
MESSAGE_SF  EQU    $00
MESSAGE_EM  EQU    $00
MESSAGE_ES  EQU    $46
MESSAGE_EF  EQU    $65

;TRACK NO.=02

;88-09-05          01:34:26                    SYSTEM_HUDSON.IPL
HUDSON_HI   EQU    $00
HUDSON_LW   EQU    $0000
HUDSON_RL   EQU    $0001
HUDSON_LN   EQU    $0800

;88-09-22          23:44:28                    _IPL.INF
IPL_HI      EQU    $00
IPL_LW      EQU    $0001
IPL_RL      EQU    $0001
IPL_LN      EQU    $0080
```


CDEMUL.EXE

Chapter 9. CDEMUL.EXE

9.1 Outline

This substitutes for the CD-ROM drive by interpreting the CD-ROM command which is sent from Hu7 through parallel I/O connected to the Hu7 system. Pressing the ESC key will return to the command line.

9.2 Execution Method

CDEMUL.EXE is executed from the MS-DOS command line.

9 3 Format

CDEMUL [Option]

9.4 Options

A	Execute player simulation
D	Display output command and monitor the output commands
Q	Return other than PLAY-STATUS by SUB-Q sense
S	Emulation without emulating SEEK TIME
--	Display directions

```
PC ENGINE CDROM DRIVE EMULATOR VERSION 1.00
Direction: CDEMUL  Option
           A       All audio tracks
           D       Display the command sent
           Q       Return other than PLAY-STATUS by SUB-Q sense
           S       Do not emulate SEEK TIME
```

9.5 Directions

We will give an example of actual use.

To simulate player

CDEMUL A

Execute with the A option. Player simulation will be executed and will be displayed from the PC engine side. Used to check music.

To monitor a command

CDEMUL D

Execute with the D option. It will monitor a command sent from the PC engine. Display as follows:

```
A>CDEMUL D
PC ENGINE CDROM DRIVE EMULATOR VERSION 1.00
Start CD emulation
(use EXC key to stop)
TEST UN.  READY00 00 00 00 00 00 s00m
READ TOC        DE 00 00 00 00 00 00 00 00 00 01020000s00m
READ TOC        DE 01 00 00 00 00 00 00 00 00 00592500s00m
READ TOC        DE 02 02 00 00 00 00 00 00 00 00564604s00m
DATA READ       08 00 10 00 01 00 s00m
DATA READ       08 00 10 00 10 00 s00m
DATA READ       08 00 2C F8 01 00 s00m
```

The D option can be turned on/off by hitting "D" on the keyboard.

When a program to run requires ATIME

CDEMUL Q

Execute with the Q option. If the emulation program is run without this option, SUB-Q sense will not return ATIME status. Specify it when you want to run a program to look at ATIME, etc. using SUB-Q sense.

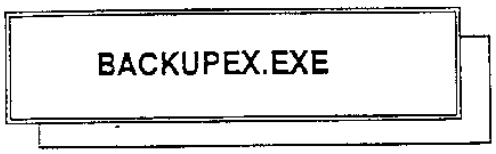
Do not emulate during CD seek time

CDEMUL S

Execute with the S option. This option does not perform during CD-seek-time, therefore, debug time will be shortened.

To end simulation

Press the ESC key to end emulation.


BACKUPEX.EXE

Chapter 10. BACKUPEX.EXE

10.1 Outline

This is to record CD Contents to 8mm MT. File control information is also recorded to the backup. A tape that is baked up by this program is used as a master.

10.2 Execution Method

BACKUPEX.EXE is executed from the MS-DOS command line.

10.3 Format

BACKUPEX Comment

10.4 Options

No options available

The following directions will be displayed if executed without any comments.

```
A>BACKUPEX
BACKUP EXA Version 1.00 Copyright 1987 Hudson soft
Directions : BACKUPEX REM
             Backup to 8mm MT according to CD_DOC.DIR
             information.
```

Example (Backup to 8mm MT with the name such as SAMPLE PROGRAM.)

```
A>BACKUPEX SAMPLE PROGRAM
BACKUP EXA Version 1.00 Copyright 1987 Hudson soft
..................................... Display a file list
```


VERIFYEX.EXE

Chapter 11. VERIFYEX.EXE

11.1 Outline

This is to compare 8mm MT contents with CD Contents. Always use this program to compare and check a master tape.

11.2 Execution Method

VERIFYEX.EXE is executed from the MS-DOS command line.

11.3 Format

VERIFYEX [File name]

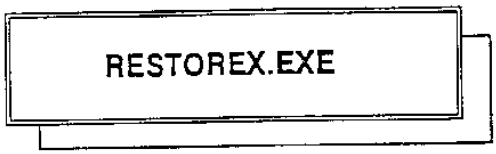
11.4 Options

It will create a file of the verified results if a file name is specified. Print this as a part of master presentation record.

Example

```
A>VERIFYEX
VERIFY EXA Version 1.00 Copyright 1987 Hudson soft
------------------------------------------------------------
HUDSON CDROM DEVELOPMENT SYSTEM ver 1.00
8mm Backup Method 1
1. This file (256 bytes)
2. HD management file header (32 bytes)
3. Data (4000H*n + 800H*m)
Repeat 2. and 3.
FMK 1 pc.
-------------------
DATE 88/09/15
TIME 03:58:42
REM  DEMO

------------------------------------------------------------
Compare to HD.  OK?  (Y)
```

To start comparison, press the Y key. A file list will be displayed. It will be OK unless an error is indicated. Pressing other than the Y key will return to the command line without a comparison.


RESTOREX.EXE

Chapter 12. RESTOREX.EXE

12.1 Outline

This is to return files of the 8mm MT recorded by BACKUPEX.EXE to CD Contents.

12.2 Execution Method

RESTOREX.EXE is executed from the MS-DOS command line.

12.3 Format

RESTORE [Option]

12.4 Options

-I Restores from the start of CD Contents. The present CD Contents will be lost and 8mm MT contents will be stored in its place.
-A Add the 8mm MT contents to the end location of current file management.

The following directions will be displayed if nothing is specified.

```
A>RESTOREX
RESTORE EXA Version 1.00 Copyright 1987 Hudson soft
Direction : RESTOREX - Switch Character
            Restores 8mm MT contents to CD Contents
            -I Restores 8mm MT contents to top of CD Contents
            -A Returns to end location of the current file
               management
```

12.5 Directions

We will give an example of actual use.

To restore 8mm MT contents to CD Contents

RESTORE -I

Execute with the -I option.

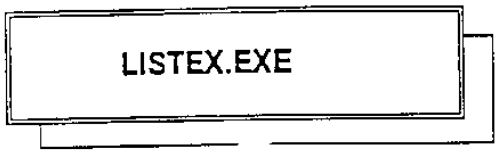
```
A>RESTOREX
RESTORE EXA Version 1.00 Copyright 1987 Hudson soft
------------------------------------------------------------
HUDSON CDROM DEVELOPMENT SYSTEM ver 1.00
8mm Backup Method 1
1. This file (256 bytes)
2. HD management file header (32 bytes)
3. Data (4000H*n + 800H*m)
Repeat 2. and 3.
FMK 1 pc.
---------------
DATE 88/09/15
TIME 03:58:42
REM  DEMO

------------------------------------------------------------
Restore to HD.  OK?  (Y)
```

Pressing other than the Y key will return to the command line.

To add 8mm MT contents to CD Contents

RESTORE -A

Execute with the -A option.


LISTEX.EXE

Chapter 13. LISTEX.EXE

13.1 Outline

This is to display a list of contents of the 8mm MT recorded by using BACKUPEX.EXE, etc.

13.2 Execution Method

LISTEX.EXE is executed from the MS-DOS command line.

13.3 Format

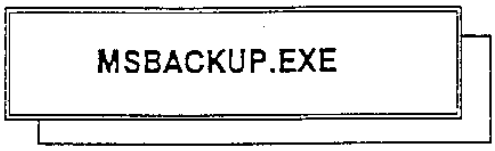
LISTEX

13.4 Options

No options available

Example

A > LISTEX
LIST EXA Version 1.00 Copyright 1987 Hudson soft
..... A file list is displayed with the header information


MSBACKUP.EXE

Chapter 14. MSBACKUP.EXE

14.1 Outline

This is to backup MS-DOS files to 8mm MT. It can also add, fetch files, and display lists.

14.2 Execution Method

MSBACKUP.EXE is executed from the MS-DOS command line.

14.3 Format

MSBACKUP [Option]

14.4 Options

/I Write a file from the start of the 8mm MT
/D Do not copy subdirectories and write directories only
/T (Date) (-Date)
Write files of the specified day or duration
/L List 8mm MT files
/R Restore from 8mm MT to MS-DOS

The following directions will be displayed if options are not specified.

```
MS-DOS 8mm MT BACKUP version 1.00
Directions: MSBACKUP [Switch] file name REM
            Copy MS-DOS files to 8mm MT
            /I E:¥*.*REM     Copy from the start of 8mm MT
            /D               Do not copy subdirectories
            /T[DATE]         Copy between specified dates
                        (88/02/12-88/04/21,today is omitted)
            /L[¥PATH]       A 8mm MT file list (/T is valid)
            /R[¥PATH]        Restore from 8mm MT to MS-DOS
                             (/T is valid)
```

14.5 Directions

We will give an example of actual use.

To backup all specified drives

```
B>MSBACKUP /I A:¥*.*
```

Explanation : Copy all in drive A: (including subdirectories) to 8mm MT.


SFTFMT.EXE

To additionally backup all specified drives

```
B>MSBACKUP B:¥*.*
```

Explanation : Copy additionally all in drive B: (including subdirectories) to 8mm MT.

To display the directory of the specified file

```
B>MSBACKUP /L¥*.ASM
```

Explanation : Display all ASM files in the root directory of the 8mm MT.

To restore the specified file to the specified drive

```
B>MSBACKUP /R¥*.ASM C:¥
```

Explanation : Copy all ASM files in the root directory of the 8mm MT to drive C:¥. The subdirectory will be automatically created.

To backup files of the specified date

```
B>MSBACKUP /T88/09/30
```

Explanation : Only backup the files dated 88/09/30 to 8mm MT.

To back up files of specified dates

```
B>MSBACKUP /T88/09/30-88/10/03
```

Explanation : Only backup the files dated between 88/09/30 and 88/10/03 to 8mm MT.

File names which can be specified by the /L and /R options are:

/L	All
/L¥*.ASM	All ASM file within root
/L¥*¥*.ASM	All ASM files in all directory
/L¥BIN¥*.EXE	All EXE files in ¥BIN
/L¥BIN¥*¥*.OBJ	All OBJ files in the directory under ¥BIN

Chapter 15. SFTFMT.EXE

15.1 Outline

This is to initialize the MS-DOS drive within CD Contents. This device driver can set a minimum of 1 drive to a maximum of 8 drives of MS-DOS partitions. Each drive can be expanded up to 60M.

15.2 Execution Method

SFTFMT.EXE is executed from the MS-DOS command line.

15.3 Format

SFTFMT [Drive : Volume name]

15.4 Options

No options available

15.5 Directions

In the following cases, use this command to format. Also backup your data before formatting.

1) When you change parameters of SCSIDEV.SYS in CONFIG.SYS.

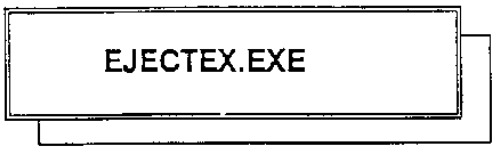
2) When you delete all the files in a temporary drive and construct a new file.

If the following drive name: volume name is input at the command line, the program will wait for a confirming Y key. If other than the Y key is pressed, it returns to the command line without doing anything. Pressing the Y key will analyze drive parameters, clear the FAT area, set a volume name at the start of a directory area, and clear everythingelse. Date and time of the occurrance will be written with the volume name.

A>SFTFMT E:60M_Disk

Format E: OK? (Y)

You can use any characters for a volume name. For the file name, only use the characters allowed in MS-DOS.


EJECTEX.EXE

Chapter 16. EJECTEX.EXE

16.1 Outline

This is to open the door of the 8mm MT unit. If a 8mm MT is installed, rewind the 8mm MT before opening the door.

16.2 Execution Method

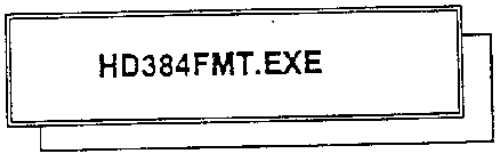
EJECTEX.EXE is executed from the MS-DOS command line.

16.3 Format

EJECT

16.4 Options

No options available


HD384FMT.EXE

Chapter 17. HD384FMT.EXE

17.1 Outline

This is to physically format the Hu7 CD Contents Hard Disk Unit.

17.2 Execution Method

HD384FMT.EXE is executed from the MS-DOS command line.

17.3 Format

HD384FMT

17.4 Options

No options available.

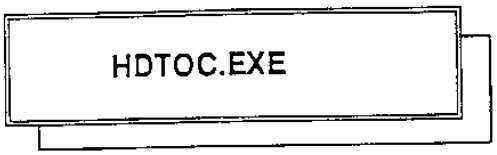
17.5 Directions

This command will bring up the following display.

```
Initialize Hu7 CD Contents Hard Disk Unit
OK?     (Y)
```

The Y key will start initialization. Other keys will end the command without initializing.

Note: It will take more than 20 minutes to execute this command. Start restoring by using the message data of the first track of a 8mm MT.


HDTOC.EXE

Chapter 18. HDTOC.EXE

18.1 Outline

This is to create files of TOC information in CD Contents. Create files of TOC information to use as a part of master documents by using this command.

18.2 Execution Method

HDTOC.EXE is executed from the MS-DOS command line.

18.3 Format

"DTOC File name

18.4 Options

No options available.

18.5 Directions

This command will output TOC information to the specified file. Print this file as a part of the documents to turn in with the masters.

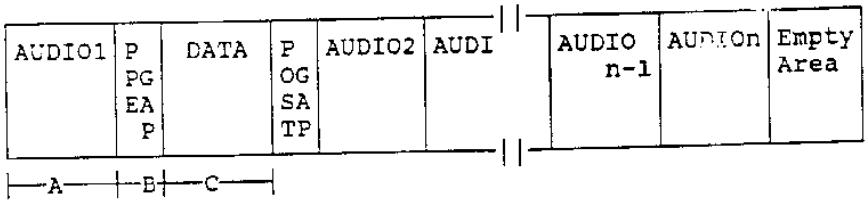
-------Information----------

Explanation of the File Management Method of CD System

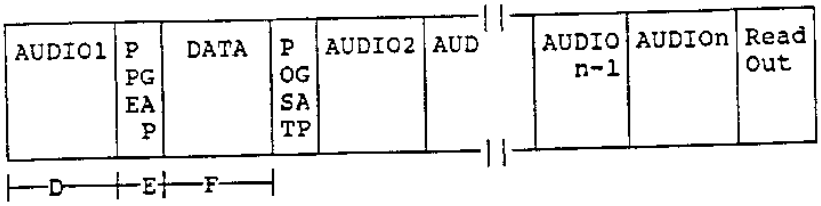
All files in CD Contents are managed by the MS-DOS files called CD_DOC.DIR. These files are located in the root directory of the first MS-DOS drive (an 8M drive can be used without option specification). This drive number is input at the execution of SCSIDEV.SYS. The drive number will not be changed later. Drive D: will remain unchanged (meaning that files may be created in a new drive). Files are not managed by MS-DOS nor a device driver. The CD_DOC.DIR file manages them in each program. Management is done in one direction and addition and deletion is possible from the end of the file as a rule. However, withn HDWRITE, if the size of a new file is smaller than or the same as the target file, it can be written over at the same location. If 8 HEX digits are specified after the /U option of HDWRITE, the capacity will be expanded from the first writing. It means that you always write on the same location. Files in the middle can be deleted by putting a delete mark on them, but it does not necessarily increase the empty area within the file. The only way to delete a file completely is to backup to 8mm MT once, then use the HDMUSIC. EXE MT command to initialize CD Contents and then restore only the specified files to CD Contents. Both DATA and AUDIO are managed by 2048 bytes per record. Only INFGET.EXE and CDEMUL.EXE are recalculated to 2353 bytes for one record. Record number 0 means 00 minutes 02 seconds 00 frame. One record is one frame (1/75 seconds).

Differences between CD system record number management and the actual CD ROM

CD system considers 2048 bytes as one record.



CD ROM (AUDIO is considered as 2352 bytes and DATA as 2048 bytes per record)



AUDIO area record computation D = (A*2048+2351)/2352 Integral value
GAP area record computation E = B
DATA area record computation F = C

CD_DOC.DIR file contents

OFFSET	LENGTH	CONTENTS	
00H	1		Attribute (80H=DATA,C0H=AUDIO,00H=Nonsense)
01H	3		Record no. of 2048 bytes unit (LOW,MID,HIGH)
04H	3		No. of record of 2048 bytes unit (LOW,MID,HIGH)
07H	6		Group name
0DH	8		File name (Label name)
15H	3		Extension
18H	4		No. of byte (From low byte to high byte)
1CH	2		Time (Time format of MS-DOS file)
1EH	2		Date (Date format of MS-DOS file)

DUMP example of CD_DOC.DIR file

```
00000000        C0 00 00 00 18 0F 00 43-44 5F 52 4F 4D 4D 45 53
00000010        53 41 47 45 00 00 00 00-00 CD 78 00 21 97 58 10

00000020        00 18 0F 90 E8 00 00 53-59 53 54 45 4D 50 52 45
00000030        5F 47 41 50 20 33 53 20-00 40 07 00 58 10 58 10

00000040        80 00 10 00 08 00 00 54-45 53 54 00 00 54 45 53
00000050        54 00 00 00 00 42 49 4E-2A 03 00 00 3D 7C 8E 10

00000060        80 08 10 00 10 00 00 54-45 53 54 00 00 46 4F 52
00000070        44 00 00 00 00 43 47 00-00 74 00 00 8C 79 8E 10
```

Above CD Contents in HDWRITE /L display

```
MD_GROUP_NAME___.EXT__BYTES_YY/MM/DD_HH:MM:SS__RECORD__LENGTH
A CD_ROM MESSAGE.    7913472 88-02-24 18:57:02 $000000 $000F18
* SYSTEM PRE_GAP.3S   475136 88-02-24 02:02:48 $000F18 $0000E8
D TEST   TEST   .BIN     810 88-04-14 15:33:58 $001000 $000008
D TEST   FORD   .CG    29696 88-04-14 15:12:24 $001008 $000010
```

About IPL

```
        seg     cseg
;-------------------------------------------------------------------------------
;       IPLINF
;
iplinftop:
        db      0       ;00 IPLBLK H    ;load start block no. of CD
        db      0       ;01 IPLBLK M    ;load start block no. of CD
        db      0       ;02 IPLBLK L    ;load start block no. of CD
        db      0       ;03 IPLBLN      ;load block length of CD
        db      0       ;04 IPLSTA L    ;program load address L
        db      0       ;05 IPLSTA H    ;program load address H
        db      0       ;06 IPLJMP L    ;program execute address offset L
        db      0       ;07 IPLJMP H    ;program execute address offset H
;
        db      ?       ;08 IPLMPR2     ;ipl set mpr2
        db      ?       ;09 IPLMPR3     ;ipl set mpr3
        db      ?       ;10 IPLMPR4     ;ipl set mpr4
        db      ?       ;11 IPLMPR5     ;ipl set mpr5
        db      ?       ;12 IPLMPR6     ;ipl set mpr6
;
        db      ?       ;13 OPENMODE    ;opening mode
                                        ;bit76543210
                                        ;   |||   |└ data read to vram
                                        ;   |||   |   0 : not read
                                        ;   |||   |   1 : read
                                        ;   |||   └── data read to adpcm ram
                                        ;   |||       0 : not read
                                        ;   |||       1 : read
                                        ;   ||└─────── bg display
                                        ;   ||        0 : display on
                                        ;   ||        1 : display off
                                        ;   |└─────── adpcm play
                                        ;   |         0 : play
                                        ;   |         1 : not play
                                        ;   └──────── adpcm play mode
                                        ;             0 : single
                                        ;             1 : repeat
;
        db      ?       ;14 GRPBLK H    ;opening graphic data record no.
        db      ?       ;15 GPRBLK M    ;opening graphic data record no.
        db      ?       ;16 GRPBLK L    ;opening graphic data record no.
        db      ?       ;17 GRPBLN      ;opening graphic data length
        db      ?       ;18 GRPADR L    ;opening graphic data read address L
        db      ?       ;19 GRPADR H    ;opening graphic data read address H
;
        db      ?       ;20 ADPBLK H    ;opening ADPCM data record no.
        db      ?       ;21 ADPBLK M    ;opening ADPCM data record no.
        db      ?       ;22 ADPBLK L    ;opening ADPCM data record no.
        db      ?       ;23 ADPBLN      ;opening ADPCM data length
        db      ?       ;24 ADPRATE     ;opening ADPCM sampling rate
;
```

```
        db      0       ;25             ;(reserve)
        db      0       ;26             ;(reserve)
        db      0       ;27             ;(reserve)
        db      0       ;28             ;(reserve)
        db      0       ;29             ;(reserve)
        db      0       ;30             ;(reserve)
        db      0       ;31             ;(reserve)
;
        db      'PC Engine CD-ROM SYSTEM',0     ;(ID string)
        db      'Copyright HUDSON SOFT / NEC Home Electronics, Ltd.',0
        db      '                       ;program name (16 bytes)
        db      '       '               ;            (6  bytes)
;=======================================================================================
```