

SEGA-CD

TECH BULLETIN

To: Sega-CD Developers
From: S. Bayless

Date: 6/16/92
CC: J. Huether
T. Reuterdaahl

Subject: The U.S. BIOS

Although you haven't seen it yet, SOJ is currently revising the BIOS for the U.S. market. While most of the changes are cosmetic, there are a couple that aren't.

Security Code

The block of code that puts up the Sega logo, prints a copyright message and plays the little "Segaaa" sound bite has been changed significantly. This has two effects. First, you'll be getting an assembly language file with a bunch of DC.B's in it which you'll need to put at the beginning of your IP before the new ROM will accept it. Second, you'll need to change the values stored in the ID Block of your boot sector.

Boot Sector

Because the security code has gotten quite a bit bigger, you'll almost certainly have to change your System ID to handle an IP larger than \$600 bytes. The "normal" System ID shown on pages 18 and 19 of the Mega-CD Disk Format Manual shows the IP and SP addresses like this:

```
          ;Disk Address,Load Size,Entry Offset,Work RAM
DC.L $00000200,$00000600,$00000000,$00000000      ;IP
DC.L $00000800,$00007800,$00000000,$00000000      ;SP
```

Note that IP starts at \$200, which is just past the ID Blocks, and SP starts at \$800 which is the beginning of Sector 1. (Remember, each sector is \$800 bytes long.) Incidentally, the \$7800 in the manual is wrong. It should be \$7200, since it's intended equal the number of bytes remaining between the end of IP and the end of sector 15.

With the new security code, IP spills over into Sector 2 and the new System Area structure (see page 6 of the Mega-CD Disk Format Manual) looks like this:

\$000000	System ID	Sector #0	Boot Sector
\$000100	Disc ID		

\$000200	IP Code		
\$001000	SP Code	Sector #1~#15	SP Sector
\$008000	VD		Data Area
	Directory		
	Path Table		
	User File 0		
	User File 1		
	.		
	.		
	.		

The original BIOS seems to have been hard wired for an IP that stayed within sector 0, so you have to kind of lie to it when IP gets bigger than that. According to SOJ, the value in the IP Start field should point to the beginning of the first byte in IP *after* the first \$600 bytes. This would mean that you'd always use \$800 (since anything smaller would fit in sector 0 and you wouldn't have to mess with any of this in the first place). It also implies that IP can't extend beyond the end of sector 1, but I haven't confirmed that yet.

The value in the IP Load Size field should be equal to the size of IP - \$600. In my first attempt with this, I just padded IP out to the end of sector 1, so the length I used was \$800, which translates to "load the whole sector".

Finally, you'll need to change the SP Start field to reflect the fact that it now begins in the next sector. The value here should be \$1000, i.e. the beginning of sector 2. You may also want to change the SP Load Size field to reflect its smaller maximum size, assuming you pad the thing out like some people do.

So, the new "normal" System ID would contain entries that looked something like this:

```

;Disk Address,Load Size,Entry Offset,Work RAM
DC.L $00000800,$00000800,$00000000,$00000000 ;IP
DC.L $00001000,$00007000,$00000000,$00000000 ;SP

```



SEGA OF AMERICA, INC.
Consumer Products Division

10 pages

To: Icom Simulations
From: Dave Marshall, SOA, Tech Support
Date: February 9, 1993
Re: Pal Conversions

The following document contains information from SOJ on PAL conversion for the SEGA CD system.

If this fax is hard to read or you would like a good copy, call me at 415-802-4476.

Thanks

dave

MEGA-CD/SEGA-CD SOFTWARE PAL CONVERSION

In this technical memo, we discuss the basic requirements for conversion from NTSC to PAL or visa versa.

It should be noted that depending on the game, the game system or program procedure may differ and the following items may not apply to every game, leaving various other methods that could be applied. Also, it is important to develop the software from the ground up with PAL/NTSC conversion in mind. Conversion, once the game is completed, could channel enormous amount of time and labor.

I. Basic Items

	NTSC	PAL
MD SIDE		
MAIN-CPU CLOCK	7.67 MHz	7.60 MHz
V-INT Generation Timing	1/60 sec	1/50 sec
CD SIDE		(UNCHANGED)
SUB-CPU CLOCK	12.5 MHz	12.5 MHz ; UNCH
INT2 Generation Timing (MAIN V-INT SYNCHRONIZED)	1/60 sec	1/50 sec ; UNCH
CD-DA	44.1 KHz	44.1 KHz ; UNCH
PCM	12.5 MHz	12.5 MHz ; UNCH

II. CD-Independent Points (Points Common to the Cartridge/MEGA-CD)

- Problem: The overall game speed is decreased.

When game timing depends on V-INT, the number of program executions per second for NTSC and PAL will differ. This will cause the entire game speed to decrease.

<Measure>

Change the speed for all items - such as the sprite's moving speed, scroll speed, etc. - synchronized through INTERRUPT as follows:

Example

```
        move.w    cnt(a6),d0                ;step4
        btst.b    #06,$a10001              ;PAL(0)/NTSC(1) MAIN-CPU
        beq.s      ?ntsc                    ;
?pal:   lea        tbl_pal(pc),a0           ;Data TBL for PAL
        bra.s      ?00                     ;
?ntsc:  lea        tbl_ntsc(pc),a0         ;Data TBL for NTSC
?00:    move.l     (a0.d0.w),d0             ;
        .
        .

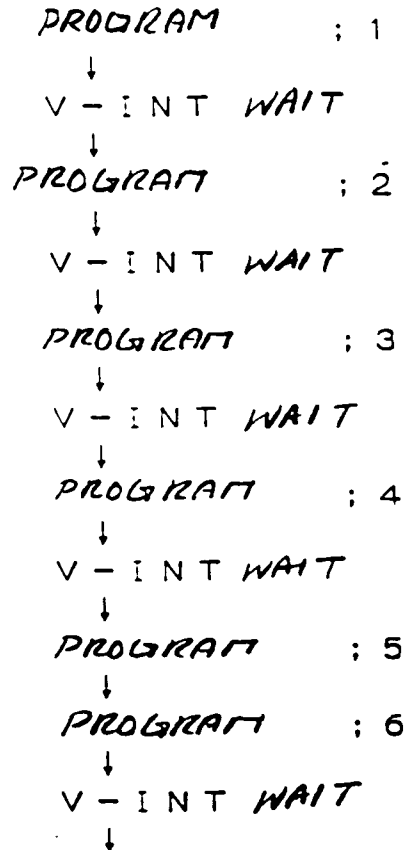
tbl_ntsc:
        dc.l      $00000000
        dc.l      $00010000
        dc.l      $00020000
        dc.l      $00030000
        .
        .

tbl_pal:
        dc.l      $00000000
        dc.l      $00013333
        dc.l      $00026666
        dc.l      $00039999
        .
        .
```

Note: Strictly speaking, this method will not create an identical item, however, the viewer would not mind the difference.

<Measure>

Only in the case of PAL, count V-INT as shown below, and execute the program once every 5 times. (Execute the program 6 times while the V-INT occurs 5 times.)



Example

```

?top:  :
        *Game Program
        btst.b  #06,$a10001          ;PAL(0)/NTSC(1) MAIN-CPU
        beq.b   ?ntsc                ;
        addq.w  #1,vint_cnt          ;
        cmpi.w  #6,vint_cnt          ;
        bne.b   ?00                  ;
        clr.w   vint_cnt              ;
        beq.b   ?top                 ;
?ntsc:  jsr     vint_wait             ;Wait V-int
        bra.s   ?top                 ;
  
```

Note 1: V-INT during the game program is not inhibited

Note 2: It is also possible to create PAL as a reference and limit the program execution to 5 times during 6 times of V-INT execution for NTSC.

- Problem: The sound is delayed.

<Measure>

When controlling sound on the MAIN side through 68000, create PAL as a reference and skip sound access once every 6 times during V-INT for NTSC.

<Measure>

When controlling sound on the MAIN side through Z80, the speed, when using FM timer for timing, for NTSC and PAL remains unchanged.

- Problem: When executing screen display in vertical 28 cell mode on the PAL monitor, the lower part of the screen (the area without any display), will flicker due to the color change during INTERRUPT.

<Measure>

After entering INTERRUPT, wait over 3 msec (or until the flicker disappears), then change color.

III. Changes and Notes which depend on the MEGA-CD

Basic Concept

The basic timing on the MEGA-CD side is 1/75 second. This timing is not effected whether the MEGA DRIVE side is PAL or NTSC. Also, the CD-ROM data playback speed, CD-DA speed, PCM playback speed are not affected by either system. Basically, as long as the MAIN-CPU side software does the processing (because of the difference between PAL and NTSC systems) correctly, no problem should occur. However, the task remains how to absorb the difference between the 2 sets of timing due to the difference in the basic operating timing.

The following example shows how data is read directly from the CD to create animation.

- Problem: Number of pixels per second.

<Measure 1>

Create an animation that creates 10 screens per second. Use 1M+1M of WORD-RAM and exchange data as follows:

Example

Using the following method, the MAIN-CPU controls timing for the screen to switch every 1/10 second.

In the case of NTSC

WORD-RAM is switched to SUB-CPU

V-INT 1	Transferring screen data to VDP
V-INT 2	Transferring screen data to VDP
V-INT 3	Transferring screen data to VDP
V-INT 4	Transferring screen data to VDP
V-INT 5	Wait
V-INT 6	WORD-RAM Return (Screen Switch)

The above total is 1/10 second

In the case of PAL

WORD-RAM is switched to SUB-CPU

V-INT 1	Transferring screen data to VDP
V-INT 2	Transferring screen data to VDP
V-INT 3	Transferring screen data to VDP
V-INT 4	Transferring screen data to VDP
V-INT 5	WORD-RAM Return (Screen Switch)

The above total is 1/10 second

<Measure>

Execute the animation depending on the speed data is read from the CD. This seems to be effective when the number of screens are less than 20 per second.

If the time required for the MAIN-CPU side to rewrite the amount of data per screen and then switch is much less than the time required for the SUB-CPU side to read the data, the timing to rewrite the screen is then controlled by reading of the more time-consuming data. This method is used equally by PAL and NTSC.

Concept Diagram

SUB-CPU	MAIN-CPU
Nth data delivery	Nth data receipt
(N+1)th screen read start	Nth screen data write start
(N+1)th screen reading	Nth screen data writing
>	>
>	>
(N+1)th screen reading	Nth screen reading completed
(N+1)th screen reading	(N+1)th screen
>	>
>	>
(N+1)th screen data delivery	(N+1)th screen

Note: Even in the case of N frames per second animation, the interval between frames will not be exactly 1/N second.

- Problem: Animation while sound is played.

<Measure>

Insert screen data between SUB side PCM (DMA) data ~ data and after a full screen of data is read, pass the data over to the MAIN side. If the MAIN side processing time is shorter than the data read time, the program is considered capable independent of the PAL/NTSC consideration.

IV. Other Necessary Changes to the CD

To create CD-ROM for PAL, security code and ID must be changed. Currently, there are 5 types of security code:

```
SECDB_JP DAT      ; JAPAN
SECDB_US DAT      ; AMERICA
SECDB_EC DAT      ; EUROPE
SECDB_JP DTU
SECDB_EC DTU
```

<*.DAT> is the smallest size for each type. In addition, for <*.DTU> to match the largest size US version security code, dummy data has been added.

<Remark>

The security code is modified by making only one line effective among the attached <IP_HEAD.ASM> defining lines:

```
; US      set      1
; EUROPE  set      1
; JAPAN   set      1
```

After assembling, linking, and creating the binary file, with respect to the disc image, it can be quickly modified through the use of <ISOUTIL.EXE>. And the process is completed by the following command:

```
isoutil d:kiscimg.emu 1 -us ipsys.bin
```

To modify only the security data, there is no need to assemble other files, build track, or build disc. Moreover, please note that to keep track of the PAL and NTSC status on the SUB side, data should be sent from the MAIN side to the SUB side.

V. Changing NTSC to PAL Conversion Specifications In the System Development

1. Changes Common to the Cartridge/CD

1) Replacing SUPER TARGET Crystal Chip
NTSC 53.693 MHz ---> PAL 53.2034 MHz

2) Changing SUPER TARGET DIP SWITCH
DIP SW4 ON; JAPAN
 OFF; USA/EUROPE

DIP SW5 ON; NTSC
 OFF; PAL

2. In the case of MEGA-CD

1) Replace BOOT-ROM on the MEGA-CD I/F board.
There are 3 versions of BOOT-ROM: JAPAN, U.S., and EUROPE per ROM)

- SUPER TARGET and MEGA-CD I/F Board
When using MEGA STATION, the above board is included in the MEGA STATION.

January 12, 1993
Sega Enterprises, Ltd.

- We currently distribute our BOOT-ROM and security code for overseas applications only to companies which manufacture overseas versions.
- For BOOT-ROM, security data, and PAL crystal chip, please contact the following address:

Attn: Asai
Planning Department I
Sega Enterprises
Ohta-ku 1-2-12
Haneda-shi, Japan 144
Tel: 03-3743-7559/Fax: 03-3745-6392

```

*****
* IP's Header(System ID & DISC ID)
*****

```

```

:US      set      1
:EUROPE  set      1
:JAPAN   set      1

```

Jan 12, 1993
Sega Enterp.

```

*system ID
*

```

```

dc.b  "0123456789ABCDEF"
dc.b  "SEGADISCSYSTEM"
dc.b  "SEGAIPMENU ".0
dc.w  $0000
dc.w  $1
dc.b  "SEGASAMPLE ".0
dc.w  $0000.0
dc.l  $800.$1000.0.0
dc.l  $1800.$2800.0.0
*SYSTEM reservation area
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "

```

```

*Disc ID
*Disc Volume name
*Volume Version
*volume type
*System name
*system Version.0

```

```

*Disc ID
*

```

```

dc.b  "0123456789ABCDEF"
dc.b  "SEGA MEGA DRIVE"
dc.b  "(C)T-12 1992.SEP"
dc.b  "SEGA SAMPLE"
dc.b  "
dc.b  "SEGA SAMPLE"
dc.b  "
dc.b  "GM T-12345 -00"
dc.b  "J"
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "
dc.b  "

```

```

*Hardware system ID
*Company code. Release date
*Game name for Japan

*Game name for Japan

*Disc kind, good number, ver no.
*I/O support

*MODEM information

```

```

ifdef  US
dc.b  "U"
include ..%inc%secdb_us.dat
else
ifdef  EUROPE
dc.b  "E"
include ..%inc%secdb_ec.dtu
else
ifdef  JAPAN
dc.b  "J"
include ..%inc%secdb.dtu
else
exit  "Not define country"
endc
endc
endc

```

```

;[
*COUNTRY      *security data for US
;][
*COUNTRY      *security data for EUROPE
;][
*COUNTRY      *security data for JAPAN
;][
;]

```



SEGA OF AMERICA, INC.
Consumer Products Division

MEGA CD TECHNICAL BULLETIN #1

To: Developers and Third Parties

From: Dave Marshall, Technical Support *DM*

Date: December 8, 1992

Re: Mega CD Technical Information

=====

The attached document contains corrections and updates to the Mega CD manual.

MEGA CD Technical Information #2

I. Corrections & Additions to the Manual

1. Manual: MEGA CD HARDWARE MANUAL

✎ Correction: PCM Sound Source, P. 11
 4-5 Micro Computer Interface Table

Incorrect

Status	External wave memory		Internal memory	
	Read	Write	Read	Write
While sounding	Impossible	Access in a period of 16 source clock cycles or more	Impossible	Access in a period of 384 source clock cycles or more (0008H)
While sounding suspended	Possible	Possible	Impossible	Access in a period of 48 source clock cycles or more (0008H)

Correct

Status	External wave memory		Internal memory	
	Read	Write	Read	Write
While sounding	Impossible	Access in a period of 18 source clock cycles or more	Impossible	Access in a period of 385 source clock cycles or more (0000H ~ 0008H)
While sounding suspended	Impossible	Impossible	Impossible	Access in a period of 49 source clock cycles or more (0000H ~ 0008H)

The above "Possible" became "Impossible" due to hardware limitations. When using MEGA-CD PCM sound, and setting the bit 7, which is sound ON/OFF bit of the Control Register (0007H), in OFF position, normal "read" and "write" to external wave memory as well as normal memory retention may not happen. As a result, the above setting should always be in ON position. Also, when stopping sound, make sure to put the stopping channel bit of the channel ON/OFF register (0008H) in OFF position.

Other Corrections:

- ☛ P61 (throughout the page) DAM → DMA
- ☛ P61 (3rd line from the bottom) WORD-RAM → WORK-RAM

2. MEGA-CD BIOS Specifications

(1) P40 BRMWRITE IN

x IN: d0.w function number
a0.1 pointer number

✓ IN: d0.w function number
d1.1 zero (0)
a0.1 pointer number

(2) Add preceding BRMINIT

How to access Back-Up Ram Cartridge

Back-Up Ram Cartridge is controlled by main CPU.

Make the BIOS ENTRY for Back-Up RAM as follows:

For back-up RAM: mburam
(For internal RAM: _buram)

Set-up data, etc. remain the same as those of the internal RAM.
For _mburam, see MEM_MAP ASS.

Check the cartridge via "BRMINIT."

cc: Cartridge is loaded
cs: d1.w 0: No RAM
1: Not formatted
2: Other formats

3. MEGA-CD Development Manual ISOUTIL P7

- ☛ Replace with attached document

II. Other Notes & Items

1. Turtle Beach Sound Stage:

You may be using Turtle Beach Sound Stage as a tool for your hard disk recording. To avoid potential damage to your disk, please observe the following:

(1) Pay special care to CONFIG. SYS SAMPLE, specially, the files:

```
BREAK=ON
BUFFERS=20
FILES=128
LASTDRIVE=E
SHELL=C:\DOS\COMMAND.COM/P/E:256
DEVICE=C:\DOS\ANSI.SYS
INSTALL=C:\DOS\FASTOPEN.EXE C: (50,25)
```

(2) AUTOEXEC BAT SAMPLE

```
@ECHO OFF
PROMPT $P$G
PATH C:\DOS:C:\VZ:C:\TOOL
SET VZDEF=C:\VZ\MI_I.DEF
MOUSE
```

(3) To avoid possible data loss, please back up contents of the hard disk, or use an IBM that contains no important data.

■ Using ISOUTIL, files can be swapped in Track Image and Disk Image.

- Track Image Swap

```
>isoutil imagefile -p -u oldfile newfile
imagefile: file created via build track
oldfile: replaceable file in the track
newfile: replacing new DOS file
```

- Only content of the file will be replaced not its name.
- Lower case letter --> Upper case letter conversion will not occur; please confirm the title.
- Replacing new files with larger contents with older files is not possible; please keep contents of the old files relatively large.
- Make sure to add the -p option

- Disk Image Swap

```
>isoutil    imagefile    1    -p    -u    oldfile    newfile
```

Same as Track Image, only Track No. will be added

- D/A portion can not be replaced. To replace D/A file, re-do BD.
- Because -es and -us options will be added, system area of the image file can be rewritten now.

-es	destPath	Extracting system area from Image file
-us	srcPath	Revising image file system area



SEGA OF AMERICA, INC.
Consumer Products Division

MEGA CD TECHNICAL BULLETIN #2

To: Sega Developers
From: Dave Marshall, Technical Support *DM*
Date: February 10, 1993
Re: Mega CD Technical Information

=====

The attached document contains information regarding use of the backup RAM on the SEGA-CD.

(2) Hardware Reset Specifications Comments

- MEGA-CD <J>

Change the jump address when the MEGA-DRIVE Reset Button is pressed. The current prescription is as follows:

- The program jumps to the control screen (BOOT-ROM) when the Reset Button is pressed with the development company's logo, title screen, or demo screen.
- The program jumps to the beginning of the application (development company's logo or title screen, etc.) during the game.

- How to jump to the control screen

<code>_CDPL_JMP (MAINENT.I)</code>	Jumps to the control screen when jumping to the left address. However, we should start from the CD initial.
<code>we</code> <code>_CDPL_RESET (MAINENT.I)</code>	Jumps to the control screen when resetting after the address on the left is assigned to <code>MEM_MAP.ASS RESET</code> .

*For `MEM_MAP.ASS,MAINENT.I`, see "MEGA-CD DISC 1."

- MEGA-CD/MEGA-CD<E> Specifications

When the reset button on the MEGA DRIVE <E> (or GENESIS) is pressed, the application should jump to SEGA-CD logo, causing the tray to open automatically. Therefore, when A, B, and C+START buttons are pressed concurrently, please set specifications so that the program should jump to the beginning of the application.

2. Cautionary Notes when Using the Backup RAM

When accessing the back-up RAM, strictly observe the following items:

1. Do not access beyond the assigned memory area.
2. Keep the size of a saved file under 256 blocks.

Item 1:

Phenomenon: Memory is accessed beyond the assigned work size.

Corrective Measure: Install SCRATCH RAM(s) to prevent illegal RAM access as follows:

<MAIN CPU SIDE>

WORK RAM	\$FF0000 ~ \$FF7A00
WORD RAM (1M1M)	\$200000 ~ \$217A00
WORD RAM (2M)	\$200000 ~ \$237A00

<SUB-CPU SIDE>

PRG RAM	\$006000 ~ 077A00
WORD RAM (1M1M)	\$0C0000 ~ \$0D7A00
WORD RAM (2M)	\$080000 ~ \$0B7A00

Setting values other than the above could cause the following symptoms:

- Operation stops when address checker is used.
- When installed in an area other than those above per WORD RAM, then areas beyond WORD RAM could be accessed.
- When installed in an area other than those above per PRG RAM, then WORD RAM could be accessed even though it does not exist.

Item 2

Symptom: Memory is accessed beyond the assigned work size.

When a saved file has a size above 256 blocks, it may not be deleted normally.

In addition, normal writing to the back-up RAM may only occur once.

Treatment:

- (1) Keep the size of the saved files under 255 blocks.
- (2) When the size must exceed 256 blocks, save in several files.



ALEX-KID-B2

ALEX-KID-B

However, since all files would be displayed in MEGA-DC control screen - as an option (6 files in the above example), user may delete only one of the files. Therefore:

- Do not delete saved files on the option screen.
- Do not allow loading even if one of the files is deleted.



MEGA-CD SOFTWARE
SUPPLEMENT MATERIALS

1993/1/12

SEGA ENTERPRISES, LTD.



SEGA OF AMERICA, INC.
Consumer Products Division

MEGA CD TECHNICAL BULLETIN #3

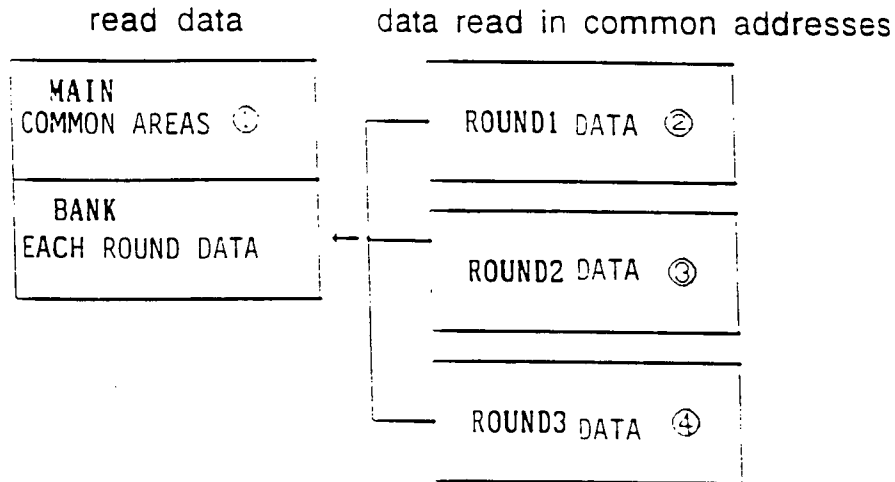
To: Sega Developers
From: Dave Marshall, Technical Support *DM*
Date: February 10, 1993
Re: Mega CD Technical Information

=====

The attached document is a supplement to the SEGA-CD manual and contains notes and updates on a variety of subjects.

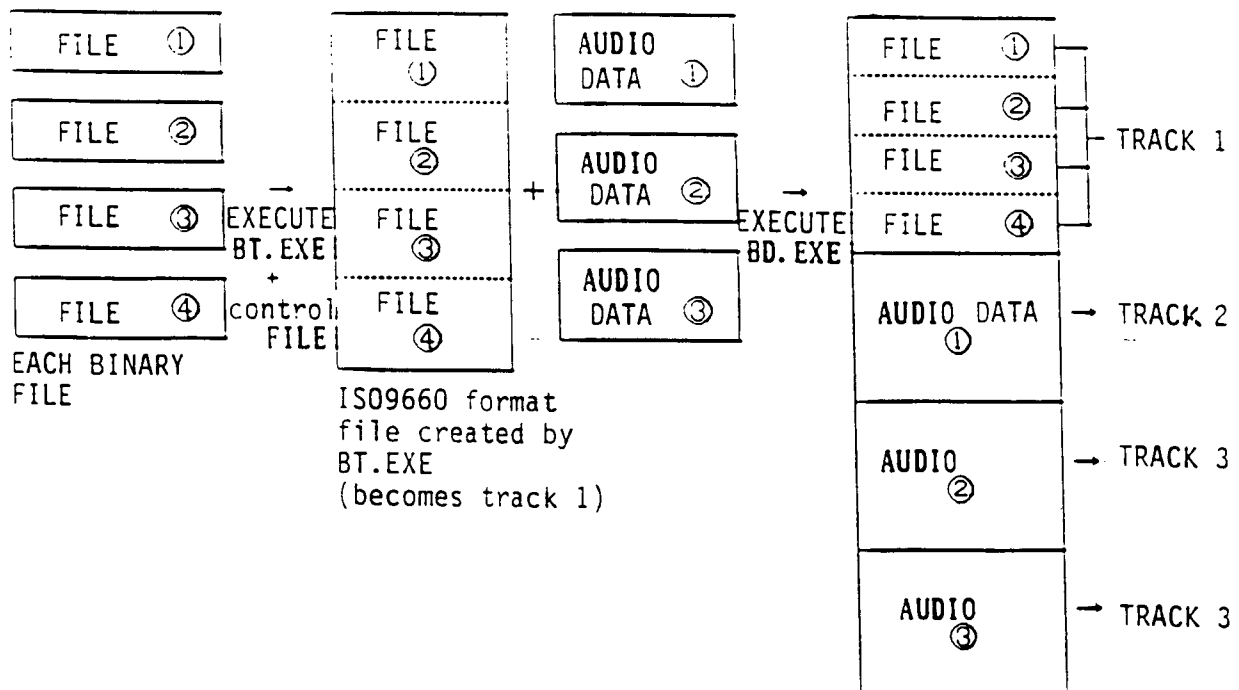
1. MEGA-CD File Configuration

What is stated here is only a basic idea: depending on the programmer, files could be created in different ways.



Procedure

- <1> Let all files 1 ~ 4 be binary.
- <2> Through BT.EXE, each file is connected to create ISO9660 file.
Using BT.EXE control file, the connecting sequence of each file is specified.
Through BT.EXE, all files 1 ~ 4 are made into the first track.
All the following tracks are CD-AUDIO.
- <3> Through BD.EXE, the first track, the second track, and the following tracks data (files that can be written once and can be executed through SEGA.EXE) are connected into one file.



2. System

PC/AT

This system is not compatible with EMS.

Use ADAPTEC for the SCSI board (for the future peripherals).

Model Name: AHA-1540B/1542B

3. Using emulator

3-1. Build Track

- BUILD TRACK control sample (see <DISC> directory)

```
Volume          ISO9660 "mega_cd"
Systemarea      "IPSYS.BIN"                ;Includes IP and SP
```

PrimaryVolume

```
SystemIdentifier      "SEGA MEGA_CD"
VolumeIdentifier      "SEGA MEGADRIIVE_GAME"
LogicalBlockSize      2048
VolumeSetIdentifier   "MEGADRIIVE_GAME"
PublisherIdentifier   "SEGA ENTERPRISES.LTD."
DataPreparerIdentifier "SEGA ENTERPRISES.LTD."
ApplicationIdentifier "SEGA ENTERPRISES.LTD."
CopyrightFileIdentifier "COPY_R.DOC"
AbstractFileIdentifier "ABS.DOC"
BibliographicFileIdentifier "BIB.DOC"
```

```
VolumeCreationDate
VolumeModificationDate
VolumeExpirationDate
VolumeEffectiveDate
```

*All the above are SEGA examples.

Lpath

Mpath

Hierarchy

```
file      'MESS.DAT'
source    MESS.COM
endfile

file      'DUMMY'
source    P.P
endfile

file      'COPY_R.DOC'
source    COPY_R.DOC
endfile

file      'ABS.DOC'
source    ABS.DOC
endfile

file      'BIB.DOC'
source    BIB.DOC
endfile
```

```
EndHierarchy
EndPrimaryVolume
EndVolume
```

Track images follow ISO9660.

	ISO9660	TEST.COM	
\$000000 (sector 0)	SYSTEM AREA	ID+IP	sector 0
In the case of sector 1		SP	sector 1-15
\$008000 (sector 16)	VD PRIMARY	VD PRIMARY	sector 16
\$008800 (sector 17)	VD TERMINATE	VD TERMINATE	sector 17
\$009000 (sector 18)	DIRECTORY	DIRECTORY	
sector 19			
	PATH TABLE	PATH TABLE	
sector 20			
	FILE	"MESS.DAT" MESS.COM	
		"DUMMY" P.P	
	FILE		

As stated in BT, SYSTEMAREA "IPSYS.BIN" file is located between sector 0 and sector 15.

(When IPSYS.BIN exceeds 32 Kbytes the replayed file can not be guaranteed.)

Files designated as:
HIERARCHY

FILE

⋮

ENDHIERARCHY

will be recorded as actual files. Therefore, "MESS.DAT" and "DUMMY" will be recorded as "MESS.COM" and "P.P" further down relative to sector 17.

3-2. Build Disc

- BUILD DISC control program sample (see <DISC> directory)

```
disc                                CDROM                                d:%disc%disc.emu
catalognumber                      00000000000000

leadin                             model
empty                             4500
postgap                           150
endtrack

track                              model
puase                             150
source                            c:%disc%mega_cd
postgap                           150
endtrack

track                              audio
puase                             150
source                            c:%disc%cdaudio0.dat
endtrack

track                              audio
puase                             150
source                            c:%disc%cdaudio1.dat
endtrack

track                              audio
puase                             150
source                            c:%disc%cdaudio2.dat
endtrack

leadout                           audio
empty                             500
endtrack

enddisc
```

To prevent noise when
playing back cd/da.
Insert 150 frames in
each track (2 seconds).

The following is a description of how to edit program data through the emulator and write onto the write-once disc.

General Rules

- All files are created on the hard disc in MS-DOS format.
 - Audio data files also must be in MS-DOS.
- (1) The created program data file is assemble-linked and binary-filed.
 - (2) BT.EXE is executed on an IBM, and the binary file is converted into ISO9660 format CD-ROM track file. At this stage, files do not include CD-AUDIO, sync and header information, etc.
 - (3) BT.EXE is executed on an IBM, and disc image files for emulation is created from ISO9660 format CD-ROM track file and audio data file. This file includes sync and header information.
 - (4) Through the execution of SEGA.EXE on an IBM, disc image file created per step 3 can be played back under the same condition as when disk is played back on the CD-DRIVE.
 - (5) Also, by setting write-once board and write-once writer on an IBM and executing WO.EXE, the file in step (3) can be written onto the write-once disc.
 - (6) By setting disc created per step (5) in development system switching box, the disc debug will also become possible.

OPTIONS when executing BUILD TRACK and BUILD DISC

Make sure the following options are added:

- . _ P WARNING occurs
 When WARNING occurs, make sure to erase it.
- . _ B 400 Setting the using buffer
 When the buffer is not set, the initialization occurs.
 Since the buffer size is small with this initialization
 value, make sure to set the value at approximately 400.

3-3. Pause Command of the Disc Builder

Pause numBlocks: Specifies the mute timing between tracks. The unit is in blocks and each block is 1/75 sec.

Specifying Pause (Example)

In seeking or playing through the track number, access the tip of index 01. The time for index 00 is fixed at 2 seconds per current DISC BUILDER (BD.EXE 1992/5/31 version).

(1) When not specifying pause (when numBlocks=0)

TNO	0 1	0 2	
INDEX	0 1	0 0	0 1
MUSIC	*****	*****	*****

↑ ↑
 *stopping playback *seek position through MSCPLAY, etc.
 position through MSCPLAY1, etc.

(2) When numBlocks=1

TNO	0 1	0 2	
INDEX	0 1	0 0	0 1
MUSIC	*****	---	*****

(3) When numBlocks=75

TNO	0 1	2	
INDEX	0 1	0	0 1
MUSIC	*****	-----	*****

(4) When numBlocks=150

TNO	0 1	0 2	
INDEX	0 1	0 0	0 1
MUSIC	*****	-----	*****

(5) When numBlocks=300

TNO	0 1	0 2	
INDEX	0 1	0 0	0 1
MUSIC	*****	-----	*****

4. Suggestions when creating MEGA-CD software

- (1) MAIN-side RAM is only SFF0000 ~ SFFFFFFF.
- (2) To match DA and picture timing, use rel_time returned by BIOS.
- (3) When reading data from CD, use PAUSE instead of MSCSTOP.
STOP will cause the disc stop rotating, causing delay when restarting.
- (4) When reading from CD, reading a number of small files is time consuming; it is preferred to read fewer files even though they might overlap.

5. PCM

- (1) Accessing PCM sound source
SUBCPU-side address is SFF0000 ~ FF3FFF (The address is in the
↓ lower byte within WORD.)
PCM sound source-side address is S0000 ~ S1FFF

*When using DMA, only the data area (SFF2000 ~ SFF3FFF) can be accessed.

This is because the following switching is allowed:

↑
When this area is 0: COMMAND AREA
When this area is 2: DATA AREA

- (2) Writing signal waveform data
When writing 1 byte data against PCM sound source, 18 clocks (SUB-CPU clock) of wait is required.
 - When using DMA to transfer ---> 18 clocks wait is executed through the hardware.
 - When using CPU to transfer ---> 18 clocks wait is executed through the software.
- (3) Also writing to the data bank of the sound being played is possible.
- (4) Restrictions when accessing PCM sound source area
Access is possible only when the sound generating bit is ON.
 - data address area: No problem with anything other than the long words.
A wait of more than 18 clocks is required.
 - command address area: No problem with anything other than the long words.
A wait of more than 385 clocks is required.

*In the case of a long word, wait may not be applied while writing data, therefore, the sound may be affected.

*Due to the hardware restriction, when using PCM via MEGA-CD, and setting the control register sound generation ON/OFF bit (bit 7) in the OFF position, READ/WRITE to the external signal (waveform) memory, as well as the memory content retention may not occur properly. Therefore, make sure to use the PCM sound source with sound generating ON/OFF bit in the ON position and to use channel when stopping sound generation.

6. Rotation Reduction

- (1) When rotation reduction operation is completed and the image buffer V dot size becomes 0 while interrupt is applied, please note that during the last line operation, the actual count will become 0 with interrupt being applied (this is explained in the hardware manual).

7. Animation

- (1) Theoretically, we can have 15 frames per second on a full screen with a capability for both FM and PCM sound sources processing - even if CD-D/A is not allowed. As the size of screen is reduced, the processing capabilities will increase. Data is read from the CD-ROM to the PRG-RAM and then transferred to the WORD-RAM. It is then transferred to the MAIN V-RAM.
- (2) The CD-DRIVE clock (16.9344 MHz) has an error margin of $\pm 1.5\%$, therefore, the number of frames per second has the same error rate. As a result, the application software must be designed to absorb $\pm 2 \sim 3\%$ error rate. This also applies to the emulator.

8. BIOS

- (1) If the INT2 on the SUB side is not applied for 100 msecs (500 msecs for RESET), BIOS may not operate correctly.
- (2) The INT2 on the SUB side does a maximum of 100 μ secs of processing.

9. Backup RAM

- (1) Backup RAM when using BIOS
(total capacity) -S40-S40-S40 BYTE can be used.
The last ID area
Directory area
Leading reserve area
- (2) When executing BIOS BRMDEL, the directory and the data is re-compressed; therefore, when interrupt occurs during delete, the file may get damaged.

10. BOOT-ROM

- (1) Reset during the game
Reset is defined in the standards for creating software. Return to the control screen, however, after pressing reset button when displaying development company's logo, title screen, or demo screen. Also return to the beginning of the application (development company LOGO, etc.) during the game play.

How to jump to control screen

_CDPL_JMP Returns to control screen by jumping to the left address.

(MAINENT.1) Please note the status (setting) at CD INITIAL.

CDPL_RESET Jumps to control screen when reset after entering (assigning) this address to MEM_MAP.ASS RESET.

Note: Please refer to BOOT ROM-related disc for MEM_MAP.ASS and MAINENT.1.

11. Sound

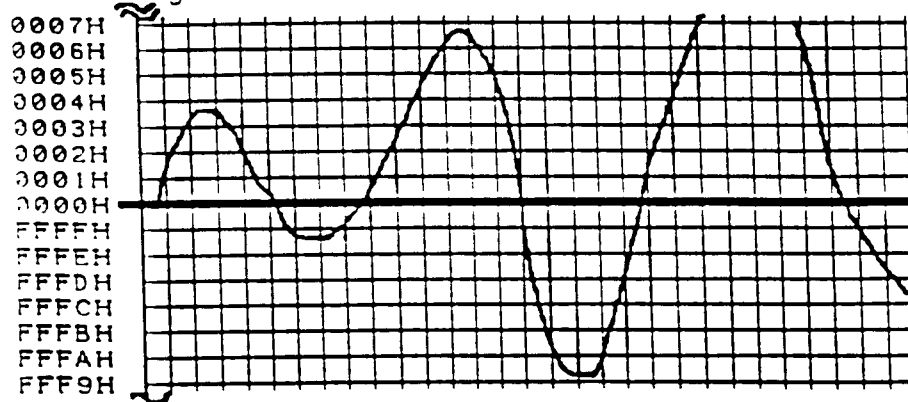
- (1) FM sound source may develop poor balance with CD-AUDIO. This could be corrected by improving balance through the MD headphone. When listening through the MEGA-CD, the MD headphone sound will mix with the CD sound; therefore, FM sound source sound volume could be adjusted through the MD headphone volume.

12. Other Items (selected form Q & A)

- (1) With MEGA-CD specifications, a minimum of 2 tracks is necessary. Enter 1 track of audio at least.
- (2) Each track requires 300 frames at least.
Use 0 dummy to fill, if necessary.

- (3) Add -S option when WARNING occurs with SEGA.EXE while using Fujitsu Write-Once Writer.
- (4) Data can not be overwritten when backup RAM is full.
Therefore,
rewrite after erasing the RAM.
- (5) The linear speed of the MEGA-CD is fixed at 1.25 m/sec. Should the using speed exceed this value, hang-up could occur as a result of the seek error.
- (6) The warning message at the beginning of the MEGA-CD can not be played back through the MEGA-CD. This is because CD-ROM can not be played back through the CD player. However, there are machines available that could play the CD ROM portion back, therefore, as a measure, the message is entered at the very beginning.
- (7) The source clock for PCM chip is Max. 12 MHz as listed in the manual, however, with MEGA-CD, the source clock is set to 12.5 MHz.

(8) Sound Stage PCM Data



As shown above, PCM increases in the direction of 0001, 0002, or decreases in the direction of FFFF, FFFE, and FFFD, with 0000H at the center.

Similarly, sampled data is recorded on the disc in the order of L-LOW, L-HIGT, R-LOW, R-HIGT, L-LOW, and L-HIGT.

Example

When: 00 10 10 F1 00 20 10 F0 00 20 10 F0 ---
 Then: Lch 1000H 2000H ---
 Rch F110H F010H ---

[GAME DISC STRUCTURE]

GAME DISC							
Read in Area		Program area				Read out Area	
CD-ROM area		CD-DA area					
00:00:00						60:04:00	
CD-DA track	CD-ROM track		CD-DA track		CD-DA track		
TNO=00		TNO=01		TNO=02-99		TNO=AA	
CD-ROM data track		Music track		Music track		Music track	
P	Data	P	P	Data	P	Data	P
00:02:00						P:Pause	
System area		Data area					
Sector #0-#15		Sector #16-#269999					
System area							
Boot sector (#0)				SP sector (#1-#15)			
Boot ID		Initial program (Security code)		System program (File system)			
System ID / Disc ID							
Data area							
#16	#17	#18-					
PVD	VDST	Path Table	Directory	File0	File1	File2...	

PVD:Primary Volume Descriptor
VDST:Volume Descriptor Set Terminator

[Memory Map]

● MEGA DRIVE SIDE

MAIN-CPU MEMORY AREA	
[1Mbit BOOT-ROM]	
\$000000	BOOT PROGRAM & DATA
\$016000	
\$01A000	CD-BIOS (COMPRESS)
\$01FFFF	DATA
[2Mbit WORD-RAM]	
\$200000	AP USE
	AP USE
\$23FFFF	
[512Kbit WORK-RAM]	
\$FF0000	IP 1.5Kbyte
\$FF0600	AP USE
\$FFF700	
\$FFFC00	ROUTINE INSIDE ROM WORK-RAM
\$FFFD00	STACK
\$FFFFFF	SYSTEM WORK-RAM

● MEGA-CD SIDE

SUB-CPU MEMORY AREA	
[1Mbit PROG-ROM]	
\$000000	CD-BIOS
\$006000	JUMP TABLE 2Kbyte
	SP
\$00xxxx	2-30Kbyte
	AP use
\$07FFFF	
[2Mbit WORD-RAM]	
\$080000	AP USE
	AP USE
\$0BFFFF	
[1Mbit WORD-RAM]	
\$0C0000	AP USE
\$0DFFFF	

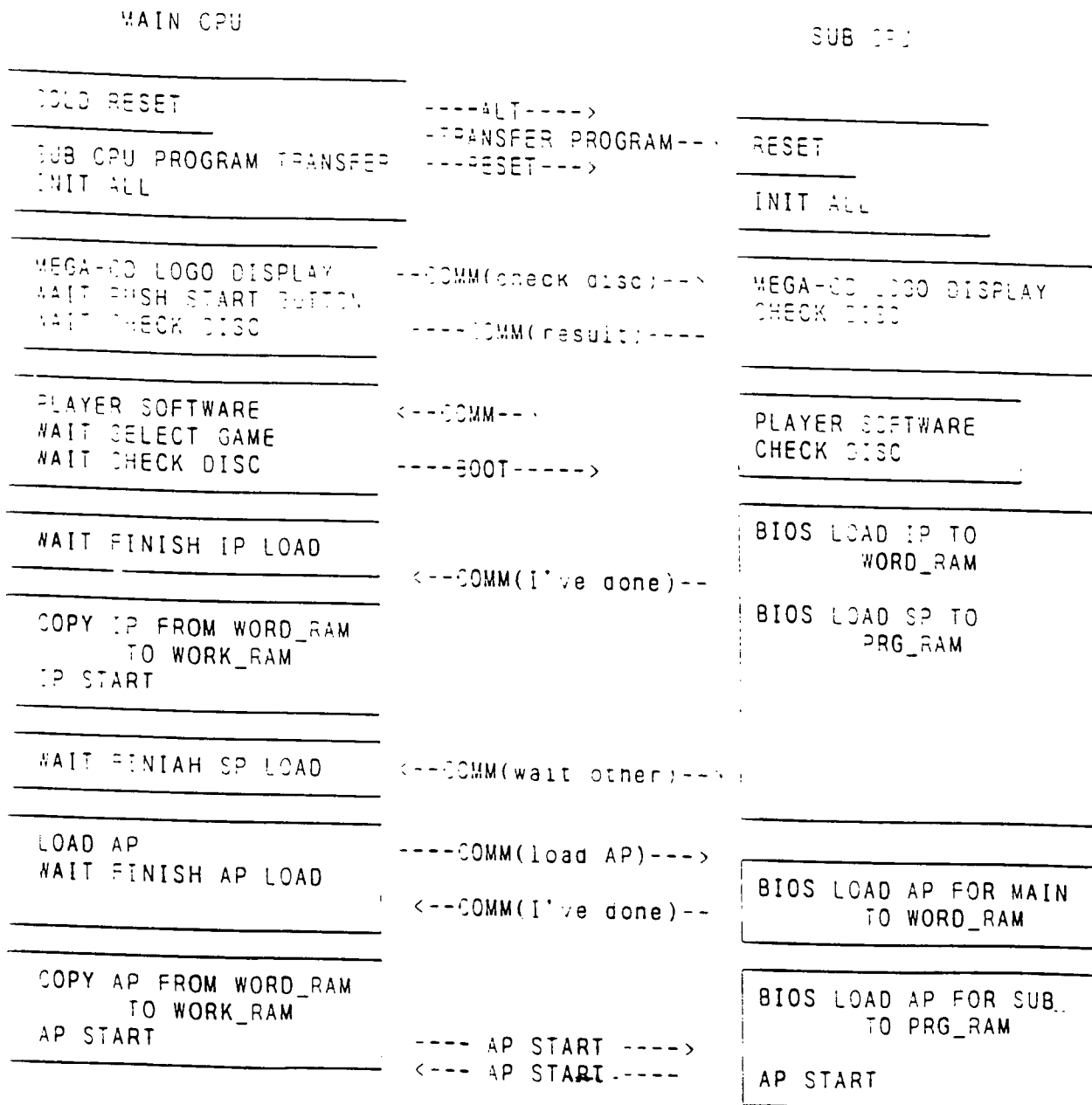
```
*****
* BOOT DIAGRAM
* *****
```

ITEMS :

- a. Loading IP
- b. Loading SP

Make sure to read notes in step 6

1. Procedure



2. Creating Boot ID

In creating boot header, attention must be paid to the start address and the length of IP.

When the IP length is less than \$600, the *1 line will be:

```
dc.b      S200, S600.0.0
```

When the IP length exceeds \$600, the value, excluding the first \$600, is entered. As an example, when the IP length, including the boot ID, is \$1800, the *1 line will be:

```
dc.b      S800, S1000.0.0
```

Also, depending on the game, the following must be modified:

```
DISC VOLUME NAME
SYSTEM NAME
GAME NAME
DISC KIND, GOOD NUMBER
```

dc.b	"DEGA01SCSYSTEM "	*Disc ID
dc.b	"DEGA1PMENU "0	*Disc Volume name
dc.w	\$0000	*Volume Version
dc.w	\$1	*Volume type
dc.b	"DEGA SAMPLE".0	*System name
dc.w	\$0000.0	*system Version.0
dc.l	\$800, \$1000.0.0	*1 *IP start.length
dc.l	\$1800, \$2800.0.0	*SP start.length
*SYSTEM reservation area		
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"
dc.b	"	"

*Disc ID	:	"0123456789A8CDEF"	
dc.b	"SEGA MEGA DRIVE "	*Hardware system ID	
dc.b	"(C)SEGA 1992.FEB"	*Company code, Release date	
dc.b	"SEGA SAMPLE PROG"	*Game name for Japan	
dc.b	"ECTION "		
dc.b	" "		
dc.b	"SEGA SAMPLE PROG"	*Game name for US	
dc.b	"ECTION "		
dc.b	" "		
dc.b	"GM T-60064 -00 "	*Disc kind, good number, ver no.	
dc.b	"J "	*I/O support	
dc.b	" "		
dc.b	" "		
dc.b	" "	*MODEM information	
dc.b	" "		
dc.b	" "		
dc.b	"J "	*COUNTRY	

3. The status after booting is completed.

• Main Side

The loading occurs from the beginning of WORK RAM. The stack is SFFFD00. The RAM's usage status is as follows. Please refer to the RAM and PORT assignments in 'MEM_MAP.ASS'. Subsequent to Sfffd00 are jump table, etc., and therefore, should not be used. Those preceding Sfffd00 could be freely used provided that boot ROM sample routine is not used. Also, when setting jump address in the jump table, jump address must be set to a value equal to the label value plus 2. The label value should not be rewritten. Subroutine in the boot ROM may also be used (please see maintent.i,rom_util.doc). Use "works" defined by the using routine. Enter SEGA-specified security program at the head of IP. Please note that Europe and the U.S. have different contents and sizes. Outside of the system area, work RAM values could differ depending on the BOOT-ROM version; also keep in mind that the work RAM values are not guaranteed.

```
movea.l #vint_exe0._mlevel6+2    #set vint jump address
#
```

reference :

```
# Ram assign      start 0ffc000h
ip_start          equ    $ffff0000          #Initial Program start
stack             equ    $fffffd00          # $100 Stack pointer

#Use by OS's Logo
hint_on_ram       equ    $ffffea00          # $100 H-int program
map_dev_work      equ    $ffffeb00          # $C00 For Decomp Map Data

bitdevwk         equ    $fffff700          # $200 use decomp data
scr_list          equ    $fffff900          # $280 Sprite list
color_ram         equ    $fffffb80          # $80 color ram copy
#Stack            equ    $fffffd00          # $100 For Stack
os_sys_ram        equ    $fffffd00          # $200 OS SYStem RAM area
#for modem        equ    $ffffff00          # 100 use ark for modem

*****< OS system ram assign >*****
# Clear only power on
#Jump Tables
_reset            equ    os_sys_ram          #6: reset jump table
_mlevel6          equ    _reset+6           #V interrupt
_mlevel4          equ    _mlevel6+6         #H interrupt
_mlevel2          equ    _mlevel4+6         #external interrupt
_mtrap00          equ    _mlevel2+6         #TRAP #00
_mtrap01          equ    _mtrap00+6
_mtrap02          equ    _mtrap01+6
_mtrap03          equ    _mtrap02+6
_mtrap04          equ    _mtrap03+6
_mtrap05          equ    _mtrap04+6
_mtrap06          equ    _mtrap05+6
_mtrap07          equ    _mtrap06+6
_mtrap08          equ    _mtrap07+6
_mtrap09          equ    _mtrap08+6
_mtrap10          equ    _mtrap09+6
_mtrap11          equ    _mtrap10+6
_mtrap12          equ    _mtrap11+6
_mtrap13          equ    _mtrap12+6
```

```

__trap14      equ    __trap13+6
__trap15      equ    __trap14+6
__onkerr      equ    __trap15+6
__adrenr      equ    __onkerr+6
__ocoderr     equ    __onkerr+6
__oliverr     equ    __ocoderr+6
__trdberr     equ    __oliverr+6
__noodcd0     equ    __trdberr+6
__noodcd1     equ    __noodcd0+6
__nsdverr     equ    __noodcd1+6
__trace       equ    __nsdverr+6
__int_ex      equ    __trace+6
:
:VDP reg data
rg0_dat      equ    __int_ex+6
rg1_dat      equ    rg0_dat+2
rg2_dat      equ    rg1_dat+2
rg3_dat      equ    rg2_dat+2
rg4_dat      equ    rg3_dat+2
rg5_dat      equ    rg4_dat+2
rg6_dat      equ    rg5_dat+2
rg7_dat      equ    rg6_dat+2
rg8_dat      equ    rg7_dat+2
rg9_dat      equ    rg8_dat+2
rg10_dat     equ    rg9_dat+2
rg11_dat     equ    rg10_dat+2
rg12_dat     equ    rg11_dat+2
rg13_dat     equ    rg12_dat+2
rg14_dat     equ    rg13_dat+2
rg15_dat     equ    rg14_dat+2
rg16_dat     equ    rg15_dat+2
rg17_dat     equ    rg16_dat+2
rg18_dat     equ    rg17_dat+2
:
:-----
:                                     *+ $b4
:                                     *2: VDP reg      #0  data
:                                     *2:                #1  data
:                                     *2:                #2
:                                     *2:                #3
:                                     *2:                #4
:                                     *2:                #5
:                                     *2:                #6
:                                     *2:                #7
:                                     *2:                #8
:                                     *2:                #9
:                                     *2:               #10
:                                     *2:               #11
:                                     *2:               #12
:                                     *2:               #13
:                                     *2:               #14
:                                     *2:               #15
:                                     *2:               #16
:                                     *2:               #17
:                                     *2:               #18
:                                     *+ $b4+$26 = $da
:

```

• Sub Side

Loading occurs from PRG-RAM \$6000. Please note that PRG-RAM, WORD-RAM, and COMMUNICATION AREA values are not fixed. Add SUB PROGRAM HEADER to the head of SUB-CPU application program (SP).

◇ SUB-PROGRAM HEADER Specifications

1)	db	'MAIN_____' , 0	:module name, flag
2)	dw	\$0100, 0	:version, type
3)	dl	0	:ptr. next module
4)	dl	__S_TEXT+__Sdata	:module size
5)	dl	__start-__L_TEXT	:start address
6)	dl	__Sdata+__Sbss	:work RAM size
7)	_start:		
	dw	usercall0-_start-	
	dw	usercall1-_start	
	dw	usercall2-_start	
	dw	usercall3-_start	
	dw	0	:end mark(zero)

- 1) Module Names. Flags
 11 character ASCII code.
 Flag is 0 fixed.
 The head of module name must be in capital letters in ASCII and 'MAIN'.
- 2) Version S0000-S0099: pre-release version
 S0100-Srelease version
 Type 0: normal type
- 3) Link Module Pointer
 S00000000: no link module
- 4) Module Size (ROM Size)
 Total number of bytes for the program code and the initialized data.
- 5) Start Address
 The relative address from ENTRY SUB-PROGRAM HEADER for setting the jump table.
- 6) WORK RAM Size
 Total number of bytes for the initialized and non-initialized data.
- 7) Data Table Setting in the Jump Table
 _usercall0: initialize routine
 _usercall1: main routine
 _usercall2: level 2 interrupt routine
 _usercall3: user-defined routine (not called from the system)

4. BIOS Call Sequence

By setting parameters in the register, and calling BIOS entry of '_cdbios', '_buram', the BIOS functions can be used.

5. User Call Sequence

CONTENT

4 entries are prepared in the jump table to call the user program (including the boot system) from the CD-SYSTEM program:
 _usercall0, _usercall1, _usercall2, _usercall3

0)_usercall0: This is the initialize routine entry and is called before the level 2 interrupt processing is allowed.

1)_usercall1: This is the main routine entry and is called by setting the '_usermode' value in d0.w register. By pressing return, this is re-called after the level 2 interrupt processing is completed.

2)_usercall 2: This is the level 2 routine entry and is called following the CD-SYSTEM processing.

3)_usercall 3: This is the user-defined routine entry and is not called from the system.

How to use

It is possible to use the SUB-PROGRAM HEADER or rewrite directly. The jump table 1 entry is 16 bytes, and the program execution is controlled through writing code. The code that we can write for the CD-SYSTEM is one of the 3 types: jmp, rts, or rte.

Example

_level13: jmp \$10000

\$4ef9	0.1
\$0001	2.3
\$0000	4.5

*Note: When writing directly, choose one of the following methods:

1) During the 6-byte rewrite, interrupt is inhibited.

2) Rewrite in the order of address > jump code.

When 'rte', etc., is written, it may be because of a non-fixed (indefinite) address written in the 2th ~ 5th byte.

6. Cautionary Notes

When IP is loaded into the work RAM, it starts from \$FFFF0000. At this time, SP is not loaded. For the SUB side to load SP, it is necessary to request from MAIN to the SUB for INT2 at every 16 milliseconds. Also, when it is necessary to synchronize MAIN and SUB, this should be take place on the application side.

MEGA-CD Sound Development

1. Sound sources that could be used with the Mega CD
 - (1) Usable sound sources
 - (2) How to use with music and sound effects
2. Generating sound through CD-D/A
 - (1) Using CD-D/A
 - (2) How to make CD-D/A data
3. Generating sound through PCM sound source
 - (1) Using PCM sound source
 - (2) How to make PCM signal (waveform) data and sound sequence data
4. Generating sound through DMA of PCM sound source
 - (1) Using DMA of PCM sound source
 - (2) How to make DMA signal data of PCM sound source and sound sequence data
5. Generating sound through FM and PSG sound sources
 - (1) Using FM and PSG sound sources
 - (2) How to make sound sequence data
6. Generating sound by synchronizing PCM, FM, and PSG sound sources
 - (1) Using PCM, FM, and PSG sound sources synchronization
 - (2) How to make sound sequence data
7. File exchange with the programmer side
8. Sega sound editor
9. IBM hard disc recording system
10. MAC hard disc recording system

1. Sound sources that could be used with MEGA-CD

(1) Usable sound sources

- I. CD-D/A
- II. PCM (8 sounds simultaneously)
- III. FM (6 sounds simultaneously, of which 1 sound can be used with D/A)
- IV. PSG (3 sounds simultaneously)

(2) How to use with music and sound effects

- I. Generating sound through CD-D/A
- II. Generating sound through PCM sound source
- III. Generating sound through DMA of PCM sound source
- IV. Generating sound through FM and PSG sound source
- V. Generating sound by synchronizing PCM, FM, and PSG sound sources

2. Generating sound through CD-D/A

(1) Using CD-D/A

The sound is outputted as regular CD audio

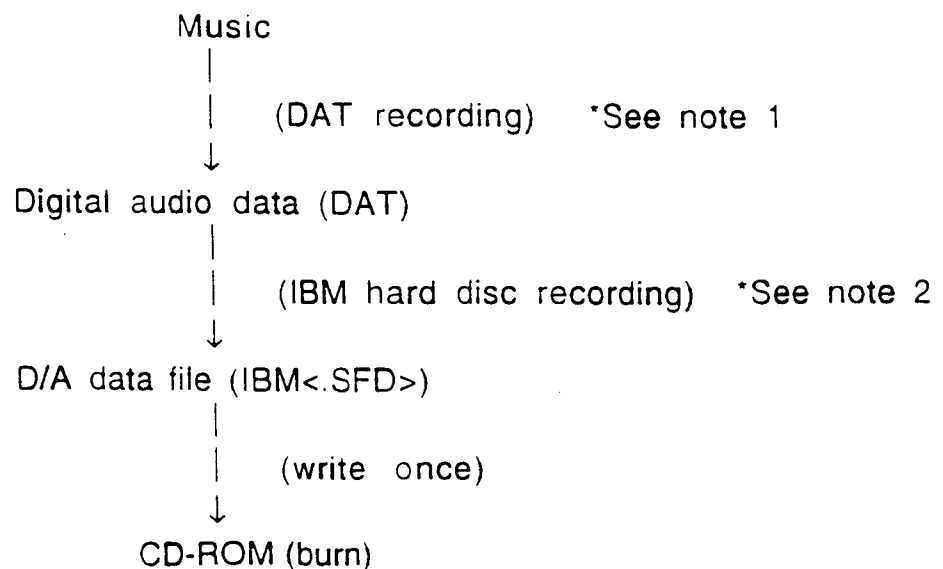
(2) How to make CD-D/A data

I. Development

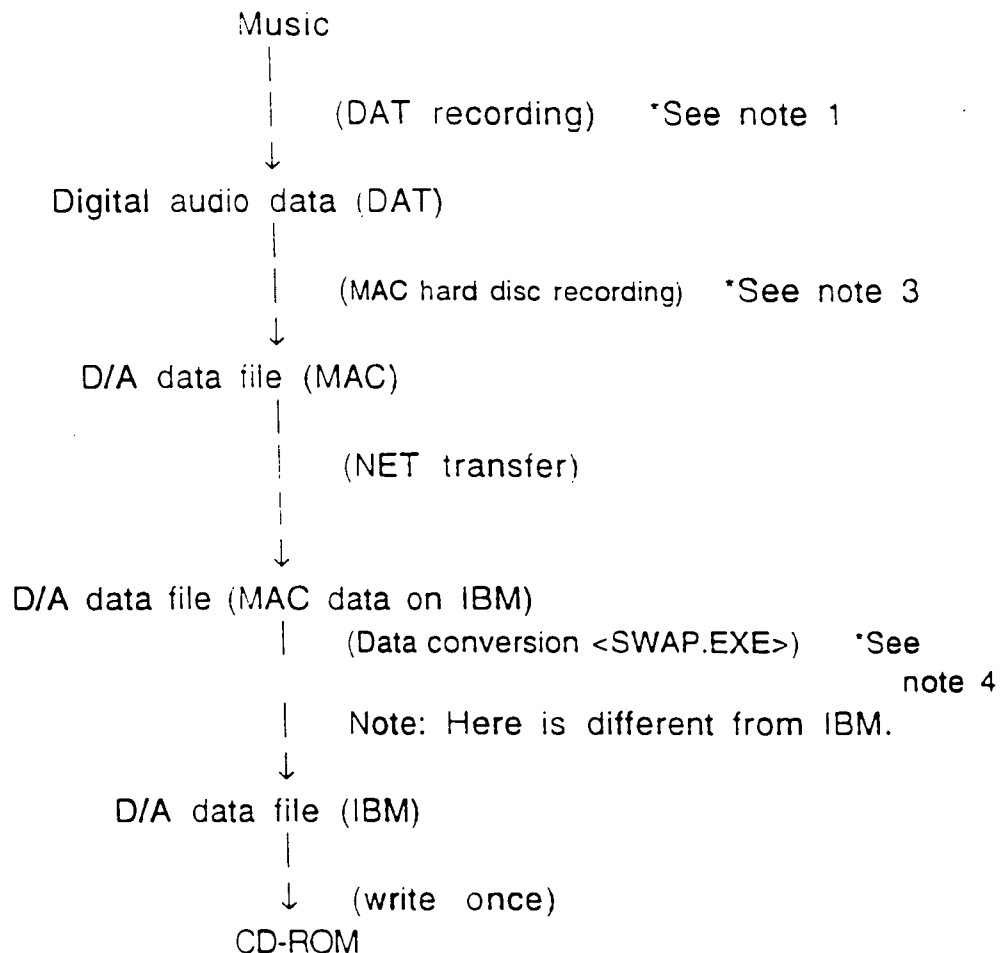
1. Record the sound onto DAT (44.1 kHz). (*See note 1.)
2. Hard disc recording (IBM, MAC).
3. Create CD emulator file. (IBM-built disc <BD.EXE>)
4. CD simulator (IBM CD emulator <SEGA. EXE>)
5. Transferring recording file as is (NET, MO disc, etc.)

II. How to create data

- Using IBM



- Using MAC



Note 1: Must be 44.1 kHz

Note 2: using TURTLE BEACH SYSTEM 56k SOUND STAGE, when using IBM. *When using MAC, Sound Tools II may also be used.* 56K as is when using IBM. For MAC, capture on IBM or 98 (DOS machine) through NET. ---> Convert

Note 3: using DIGIDESIGN SOUND TOOLS (cf: 10)

Note 4: For MAC data, the high 8 bits and the low 8 bits of the 16 bit data must be swapped (using <SWAP.EXE>)

3. Generating sound through PCM sound source

(1) Using PCM sound source

- Using PCM sound source as the instrument sound source
- Using in combination with PCM sound driver (using SUB-CPU)

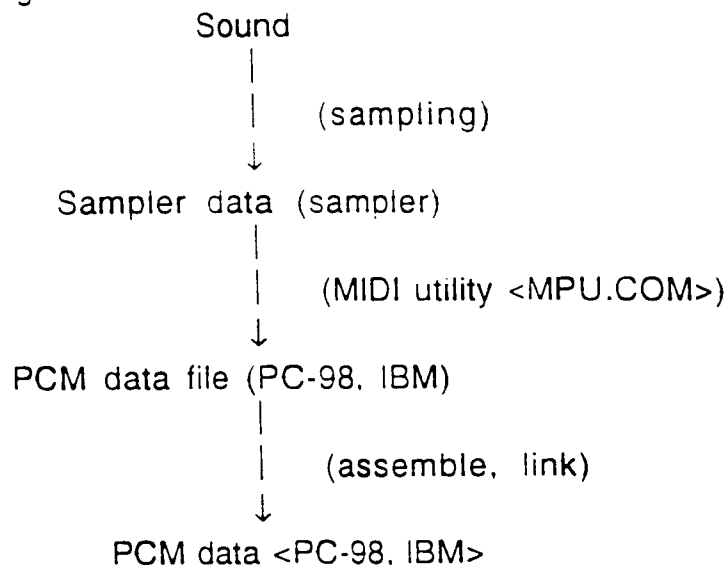
(2) How to make PCM signal data and sound sequence data

I. Development

1. Sampling by using sampler, MAC, etc.(at sound source)
2. Capturing data through MIDI (PC-98, IBM).
NET may be used if MAC and DOS are connected through NET.
3. Assemble, link, etc. - as necessary
4. Transferring as programmer file (NET, MO DISK, FLOPPY DISK, etc.)

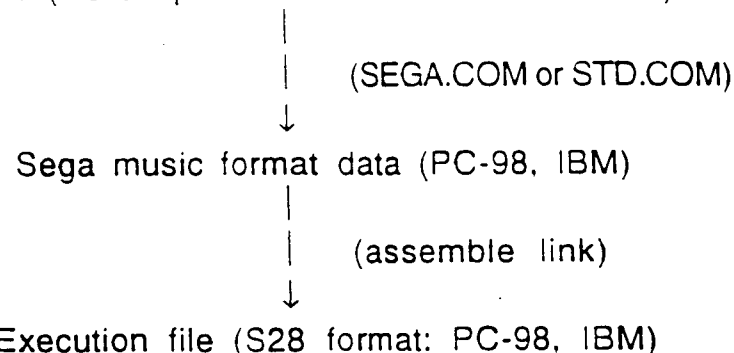
II. How to create data

1. PCM signal data



2. How to create sound sequence

Music (PC-98 prelude or standard MIDI data)



4. Generating sound through DMA of PCM sound source

(without going through the CPU - *CPU only issues data to controller*)

(1) Using DMA of PCM sound source

I. General

- Generating sound through PCM sound source while reading data from CD (recorded on CD as a file - *the same as floppy*)
- Using the combination of PCM sound driver and DMA transfer program
- A maximum of 32 kHz stereo is possible
- Transferring PCM signal data through DMA to the SUB-side RAM. (this is not a direct-signal RAM)

II. When to use

- CD-D/A can not be used
- The (total) CD capacity is not enough

(2) How to make PCM sound source DMA signal data and sound sequence data

Part 1

1. Record sound on DAT (32 kHz)
 2. Hard disc recording (IBM, MAC)
 3. Transferring file after converting to DMA (<SFD2RFD.EXE>, etc.)
- Note: <SFD2RFD.EXE> will be release in the near future.

Part 2

1. Sampling through the use of sampler, MAC, etc.
2. Capturing data via MIDI (PC-98, IBM)
3. Assemble, link, debug, etc. - as necessary
4. Transferring data after converting to binary. (floppy disk, etc.)

DMA Data

Data used as DMA could be of many formats. With <SFD2RFD.EXE>, the IBM recording data is connected to L/R 800H blocks alternately, and the recall format loop data is assigned to the specified position - thus converting to data with 0 filling its end for size adjustment.



II. How to create data

1. PCM-DMA signal data

IBM D/A data file (IBM)
(So far the same as CD-D/A)

(utility <SFD2RFD.EXE>, etc.)

PCM-DMA data file (IBM)
(L/R RECOH format binary files (2 files): <.DAL><.DAR>)

PCM data file (PC-98, IBM)
(So far the same as PCM sound source)

PCM-DMA data file (PC-98, IBM)

2. How to create sound sequence data

Sega music format data (PC-98, IBM)

(assemble, link)

Execution file (PC-98, IBM)

5. Generating sound through FM and PSG sound sources

(1) Using FM and PSG sound sources

I. General

- Using FM and PSG sound sources as an instrument's sound source
- Using the combination of FM and PSG sound drive

II. Cautionary Notes

- The MEGA-CD-side RAM can not be read from Z80. Therefore, the data should either be left in Z80 SRAM or it should be captured when exchanged with MAIN-CPU.
- PSG sound source can not be accessed from Z80. Therefore, to use PSG sound source, execute from MAIN-CPU.

(2) How to create sound sequence data

I. Development

• Music

1. Composing with sequencer
2. Converting data to SEGA music format (PC-98, IBM)

3. Integrating data into sound file using text editor
 4. Transferring to assemble, link, target
 5. Editing with sound editor (track, tone data, etc.) <target>
 6. Writing correction data into sound file
 7. Repeating steps 4 through 6
 8. Transferring after converting to programmer file (DB format).
(NET, MO disk, floppy disk, etc.)
- Sound Effect
 1. Writing data to the sound file using text editor (PC-98, IBM)
 2. Transferring to assemble, link, and target
 3. Editing with sound editor (target)
 4. Repeating steps 1 through 3
 5. Transferring after converting to programmer file (DB format).
(NET, MO disk, floppy disk, etc.)

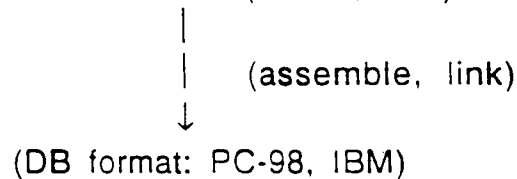
II. How to create data

1. Music

Music (PC-98 prelude or standard MIDI data)

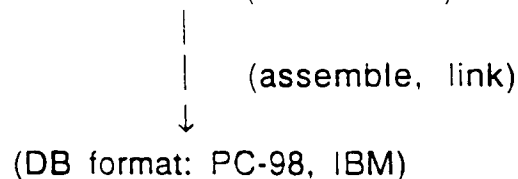


Sega music format data (PC-98, IBM)



2. Sound effect

Sega music format data (PC-98, IBM)



6. Generating sound by synchronizing PCM, FM, and PSG sound sources

(1) Using PCM, FM, and PSG sound sources synchronization

I. General

- Playing sound through Vint by synchronizing Z80 and SUB-CPU
- Using the combination of PCM sound driver (using SUBCPU), FM, sound driver (using Z80 CPU)

II. Cautionary Notes

- The MEGA-CD-side RAM can not be read from Z80. Therefore, the data should either be left in Z80 SRAM or it should be captured when exchanged with MAIN-CPU.
- Do not bus-request Z80 for a prolonged period when the music is playing.

(2) How to create sound sequence data

I. Development

1. Composing with sequencer
2. Converting data to SEGA music format (PC-98, IBM)
3. Integrating data into sound file using text editor (separating to PCM-side, FM and PSG-side)
4. Transferring to assemble, link, target
5. Editing with sound editor (music, tone data, etc.)
6. Writing correction data into sound file
7. Repeating steps 4 through 6
8. Transferring after converting to programmer file. (NET, MO disk, floppy disk, etc.)

II. How to create data

Music (PC-98 prelude or standard MIDI data)

(SEGA.COM or STD.COM)



Sega music format data (separating to PCM-side and FM-side)



(assemble link)

Execution file (PCM-side, FM-side)

7. File exchange with the programmer side

(1) CD-D/A

Transferring recorded D/A file as is

(2) Regular PCM

Transferring the created programmer file (S28 format)

(3) PCM-DMA

Transferring PCM-DMA data file

(4) FM, PSG

Transferring the created programmer file (DB format)

8. Sega sound editor

(1) Object

- Editing sound data
- Replugging super target BOOT-ROM

(2) What is currently possible?

- Data reception

PCM, FM files (discriminates files)

- PCM edit

Music data, sound effect sound data, PCM header

- FM, PSG edit

Music data, sound effect data, tone data, WORK RAM

- DC play

Play, pause, stop, open, close, music selection

9. IBM hard disc recording system

(1) Currently used system:

- 56K (digital I/F board)
- SOUND STAGE (signal edit software)

Note: both from TURTLE BEACH SYSTEM

(2) What can be done

- Capturing analog data (DAT, CD player)

Input types: 48 kHz/44.1 kHz/32 kHz

- Signal edit

Cut/copy/ volume adjustment /equalizing/mix

(3) Creating file

- Information (.SFI)
- D/A data (.SFD)

Note 1: The format is 16-bit stereo binary. Also, it can be used as direct CD simulator.

Note 2: When using SOUND STAGE, please do not put other files on recording hard disc (the file may get damaged).

10. MAC hard disc recording system

(1) Currently used system:

- SOUND TOOLS (hard disc recording system)

Note: from DIGIDESIGN

(2) What can be done

- Capturing digital data (DAT, CD player)

Input types: 48 kHz/44.1 kHz/32 kHz

- Capturing analog data

- Signal edit

Cut/past/fade/reverse/cross-fade/loop/mix/effect

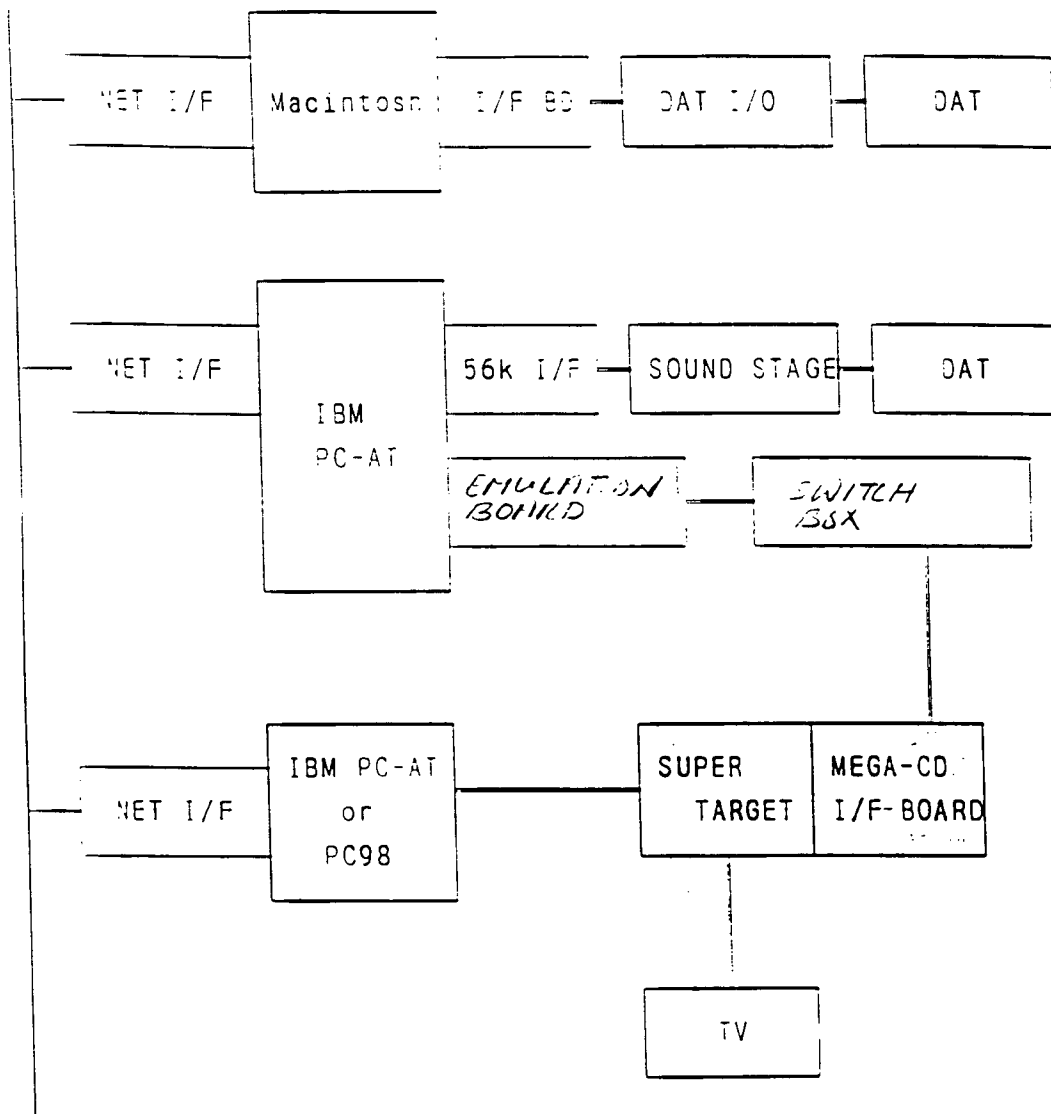
(3)Creating file

- D/A data

Note: The format is 16-bit stereo binary. However, compared to IBM, the high 8 bits and the low 8 bits are switched, therefore, to use it as CD emulator, the data must be converted (<SWAP.EXE>).

● SEGA SOUND DEVELOPMENT

NET



SEGA PRODUCT (currently sold by SI Electronics)



SEGA OF AMERICA, INC.
Consumer Products Division

MEGA CD TECHNICAL BULLETIN #4

To: Sega Developers
From: Dave Marshall, Technical Support
Date: March 15, 1993
Re: Mega CD Technical Information

=====

This technical bulletin contains updates on the basic sample playback frequency for the Sega-CD PCM sound chip.



SEGA OF AMERICA, INC.
Consumer Products Division

Appendix 1 of the Sega-CD Hardware section contains information on the PCM sound chip. Page one states that the sample frequency is 31.3 KHz. This would be the correct frequency for a chip driven by a 12 MHz clock, but the Sega-CD uses a 12.5 MHz clock. Therefore, the correct basic sample playback frequency is 32.604 KHz.

The two 8-bit registers (FDH FDL) used by the PCM chip to set the sample playback rate for a particular channel can be thought of as a single 16-bit value which can be calculated as follows:

Given: $F ::=$ base playback frequency (32604 Hz)
 $P ::=$ desired playback frequency (also in Hz)
 $M ::=$ multiplier (the value placed into FDH FDL)

Then:

$$M = P * 2048 \div F$$

Example:

If you want a playback rate of 30720 Hz,

$$M = 30720 * 2048 \div 32604$$

$$M = 1929.657711$$

Rounding this answer to 1930 and converting to hex, we get \$078A. The registers would then be set as follows:

$$FDH = \$07 \qquad FDL = \$8A$$



SEGA OF AMERICA, INC.
Consumer Products Division

MEGA CD TECHNICAL BULLETIN #5

To: Sega Developers
From: Dave Marshall, Technical Support
Date: April 5, 1993
Re: Mega CD Technical Information

=====

The attached document contains information on waiting during VRAM reading and also a problem with the Super Target Development Systems.

- o WAIT during Mega Drive VRAM READ.

In the Mega Drive, during VRAM READ, when the following conditions are not met, data may not be read correctly.

During display period and H-blank

VRAM Address Set

—
└─
VRAM READ

—
— WAIT exceeding 116 CPU clock cycles
└─

Next VRAM Address Set

During V-blank

VRAM Address Set

—
└─
VRAM READ

—
— Wait exceeding 12 CPU clock cycles
└─

Next VRAM Address Set

As seen above, after the last VRAM READ is completed, WAIT is required before the next VRAM address is set. With respect to VRAM READ, it can be read continuously.

- o SUPER TARGET IC 19

When 68000 CPU accesses an illegal area, "DATA ACKNOWLEDGE" signal may not be sent back - causing "hang up" in the production unit. Depending on some ROM emulators or ICE (+SUPER TARGET), even the emulator may hang up. To avoid such hang ups, measures have been taken so that SUPER TARGET would return ACKNOWLEDGE signal to the CPU without system hang ups. This is done via IC 19.

As a result, due to this IC function, for SUPER TARGET to function normally, the hang up occurs in the production unit.

Please take the following steps to correct the problem:

- (1) Before checking the software through SUPER TARGET + CPU, remove IC 19 from the socket.
- (2) When using ICE (ROM emulator), IC19 may be removed. Reinstall the IC should ICE (ROM emulator) act abnormally.
- (3) The above measures are common with the MEGA DRIVE and the MEGA CD.



SEGA OF AMERICA, INC.
Consumer Products Division

MEGA CD TECHNICAL BULLETIN #6

To: Sega Developers
From: Dave Marshall, Technical Support
Date: April 21, 1993
Re: Mega CD Technical Information

=====

This technical bulletin contains information on problems encountered when using Macintosh Sound Designer to produce RedBook audio for the Sega-CD



SEGA OF AMERICA, INC.
Consumer Products Division

In order to get sound out of the Sega-CD, the RedBook sound file must be in Intel (low byte, high byte) format.

Sound Designer (and other Mac sound tools) produce RedBook sound files in Motorola (high byte, low byte) format. If this file is used, the low byte gets played and produces only white noise.

This problem can be corrected by using the MAKERED utility. This utility takes a Sound Design file and converts it into a RedBook, Intel format file which can be properly used by the Sega-CD.

This tool is available in the CD-ROM conference on the Sega BBS.

If Sound Designer is also being used to produce PCM files, another issue must be addressed. The Motorola format, Sound Designer file contains SIGNED data. PCM files must be unsigned. Therefore, the files must be converted to unsigned data in order to be used for PCM sound. Since PCM data is only 8 bits, the MakeRed utility would not be needed.

MEGA CD TECHNICAL BULLETIN #7

To: Sega Developers
From: Dave Marshall, Technical Support
Date: June 8, 1993
Re: Mega CD Technical Information

=====

New standards have been adopted for the SEGA-CD ID TABLE. These standards will become effective on July 1st 1993. Following this date, the test department will be comparing the ID table against the new standards.

The following document contains the new standards for the SEGA-CD ID TABLE.



System ID

The following is an addendum to the SEGA-CD software standards, and should be used in conjunction with the *System ID* and *Disk ID* sections of the **Mega-CD Disc Format Specification** document in your CD-Rom Manual. This document is subject to change in the future.

File ID blocks (System ID and Disk ID) on all SEGA-CD programs must incorporate the following:

System ID

NO.	ADD.	SAMPLE	FIELD NAME	SIZE
1.	\$000 dc.b	'SEGADISCSYSTEM'	:DISC IDENTIFIER	16
2.	\$010 dc.b	'JURASSIC',0	:VOLUME NAME, 0 terminate	12
3.	\$01C dc.w	\$01xx	:VOLUME VERSION	2
4.	\$01E dc.w	\$0001	:VOLUME TYPE	2
5.	\$020 dc.b	'SEGAOS',0	:SYSTEM NAME,0 terminate	12
6.	\$02C dc.w	\$0001	:SYSTEM VERSION	2
Ø.	\$02E dc.w	\$0000	:ALWAYS 0	2
7.	\$030 dc.l	\$00000800	:IP ADDRESS	4
8.	\$034 dc.l	\$00000800	:IP LOAD SIZE	4
9.	\$038 dc.l	\$00000000	:IP ENTRY ADDRESS	4
10.	\$03C dc.l	\$00000000	:IP WORK RAM SIZE	4
11.	\$040 dc.l	\$00001000	:SP ADDRESS	4
12.	\$044 dc.l	\$00007000	:SPLOAD SIZE	4
13.	\$048 dc.l	\$00000000	:SP ENTRY ADDRESS	4
14.	\$04C dc.l	\$00000000	:SP WORK RAM SIZE	4
15.	\$050 dc.b	'03011993'	:DATE EMU FILE IS BUILT	16
	\$060 dcb.b	160,\$20	:SYSTEM RESERVATION AREA	160

Field Name Descriptions

- DISC IDENTIFIER**
 SEGADISCSYSTEM :For SEGA system disc.
 SEGADATADISC :For SEGA data disc

2. **VOLUME NAME**
Disc Name
3. **VOLUME VERSION**

\$00xx for samples

\$01xx for first release

;\$0000 - \$0099 (last two numbers increment by one each time new build is sent to SEGA)

;\$0100- for first rev (second rev would be \$0200, etc.)
4. **VOLUME TYPE**
\$0001
5. **SYSTEM NAME**
Developers operating system filename, cdsys.bin, 0 terminate.
6. **SYSTEM VERSION**
Developers operating system version, \$xxxx
15. **DATE EMU FILE IS BUILT**
mmddyyyy(03011993), date emu file was built. To allow easy identification of disc's for testing purposes. Fill all unused bytes with spaces, \$20.
- Ø. **ALWAYS 0**

Example

```

;-----
; BOOT HEADER
;-----
org $0                                ;header goes at beginning of memory

```

System ID:

1.	dc.b	'SEGADISCSYSTEM'	;Disk type ID
2.	dc.b	'JURASSIC', 0	;Volume ID
3.	dc.w	\$01xx	;Volume version vv.rr
4.	dc.w	\$0001	;CD-ROM = \$0001
5.	dc.b	'SEGAOS', 0	;System name
6.	dc.w	\$0001	;System version vv.rr
Ø.	dc.w	\$0000	;always 0
7.	dc.l	\$00000800	;IP disk address
8.	dc.l	\$00000800	;IP load size
9.	dc.l	\$00000000	;IP entry offset
10.	dc.l	\$00000000	;IP work RAM size
11.	dc.l	\$00001000	;SP disk address
12.	dc.l	\$00007000	;SP load size
13.	dc.l	\$00000000	;SP entry offset
14.	dc.l	\$00000000	;SP work RAM size
15.	dc.b	'03011993'	;date EMU file is built
.	dcb.b	160, \$20	;reserved





Disc ID

Disc ID

NO.	ADD.	SAMPLE	FIELD NAME	SIZE
16.	\$100 dc.b	`SEGA GENESIS`	:HARDWARE DESIGNATION	16
17.	\$110 dc.b	`(C)SEGA 1993.MAR`	:COMPANY CODE, DATE	16
18.	\$120 dc.b	`JURASSIC PARK`	:DOMESTIC TITLE	48
19.	\$150 dc.b	`JURASSIC PARK`	:OVERSEAS TITLE	48
20.	\$180 dc.b	`GM MK-XXXX -00`	:DISC TYPE, PRODUCT NO., VER.	16
21.	\$190 dc.b	`J`	:I/O DEVICE	16
¥.	\$1A0 dc.b	`	:SPACE	28
22.	\$1BC dc.b	`	:MODEM INFO.	12
¥.	\$1C8 dc.b	`	:SPACE	40
23.	\$1FO dc.b	`U`	:REGION	16

Field Name Descriptions

16. **HARDWARE DESIGNATION**
 SEGA GENESIS :For GENESIS.
 SEGA MEGA DRIVE :For MEGA DRIVE
17. **COMPANY CODE, RELEASE DATE**
 (C) **SEGA 1993.JAN** :For SEGA Developers.
 (C) **T-XXX 1993.JAN** :For third party developers
18. **DOMESTIC TITLE**
 Title for JAPAN
19. **OVERSEAS TITLE**
 Title for USA and/or EUROPE

20. DISC TYPE, PRODUCT NO., VER.
"GM" (GAME), "AI" (EDUCATIONAL).

It should be:

GM MK-xxxx -00	:For SEGA developers first rev.
GM MK-xxxx -0x	:For SEGA developers running rev.
GM T-xxxx -00	:For third party developers first rev.
GM T-xxxx -0x	:For third party developers running rev.

21. I/O DEVICE

MASTER SYSTEM JOYSTICK	:O
MEGA DRIVE(GENESIS) JOYSTICK	:J
KEYBOARD	:K
SERIAL I/O (RS232C)	:R
PRINTER	:P
TABLET	:T
TRACKBALL	:B
PADDLE CONTROLLER	:V
ANALOG JOYSTICK	:A
MOUSE	:M
6-BUTTON PAD	:6
MULTI TAP	:4
MENACER	:G
ACTIVATOR	:L

22. MODEM INFO
Not used.

23. REGION
- | | |
|--------|----|
| JAPAN | :J |
| USA | :U |
| EUROPE | :E |

- ¥. SPACE
Prohibited to use.

Note: All x's MUST be replaced with the appropriate information. For any open addresses, fill in with the ASCII code-equivalent of "SPACE" (20H). All text in **BOLD COURIER** MUST be entered as they appear in this document.

EVERY SEGA CD ROM DISC MUST CONTAIN AT LEAST 2 TRACKS.

Example



Disc ID:

16.	dc.b	'SEGA GENESIS'
17.	dc.b	'(C)SEGA 1993.MAR'
18.	dc.b	'JURASSIC PARK'
19.	dc.b	'JURASSIC PARK'
20.	dc.b	'GM MK-XXXX -00'
21.	dc.b	'J'
¥.	dc.b	'
22.	dc.b	'
¥.	dc.b	'
23.	dc.b	'U'

Additional Copies

To obtain additional copies of
this document, contact either:

Jeff Feltman at extension 4493,
or
Jeff Wagner at extention 1325.

MEGA CD TECHNICAL BULLETIN #8

To: Sega Developers
From: Dave Marshall, Technical Support
Date: June 21, 1993
Re: Mega CD Technical Information

=====

The following document contains information on the Sega-CD. Most of the information addresses different issues with the Sega-CD 2.

The tech note make references to MEGA-CD Disks. These disks contain code and notes from SOJ. These disks are currently being translated and will be available on the BBS when the translation is finished.

This tech note also makes references to a Pal conversion document. This Pal conversion document has been included with this tech note.

I. Changing Points Relating to MEGA -CD 2

1. Hardware Method

- 1) Change BOOT-ROM. (Change of opening control screen)
- 2) The CD tray's opening method changes from automatic to manual.

2. Software Method

1) Relating to BIOS

Through the method of exhibiting each BIOS call, it is completely compatible. However, there is a BIOS method change as shown below that is caused by the method change the above 1 - 2), so be cautious.

In the case where "DRVINIT" "DRVOPEN" is called, the status will not change until the open/close operation is actually performed manually.

- Avoid performing the time out function after estimating the front loading.
- The error function corresponding to manual open is the same as if a fatal error occurred in CD1
- When it becomes necessary to change a CD during a game that has two-board group disks, cope with it by doing things like using a message to request a disk change since the application cannot open the tray while in CD2. It is also not necessary to return to the control screen to change the CD again

3. Changing Point of MEGA DRIVE II (Reference)

- 1) Modem connector is gone
- 2) Video connector*source connector's shapes are changed
- 3) Stereo sound comes from the video connector (uses cable that must be purchased separately)

4. Methods to Distinguish of MEGA-CD and MEGA-CD2

- 1) BOOT-ROM's 100 H to 1 FFH contains the same type of ID code as the cartridge ROM. In the address 18CH is a ROM version and it is by this version that you can distinguish between CD and CD2.

2) CD 1's ID Code for Japan Use

dc. b	'SEGA MEGA DRIVE	'	;100:
dc. b	'(C) SEGA 1992. MAY	'	;110: release year, month, day
dc. b	'MEGA-CD BOOT ROM	'	;120: domestic title name
dc. b	' 05/01-1992	'	;130:
dc. b	'02:30 1.11	'	;140:
dc. b	'MEGA-CD BOOT ROM	'	;150: overseas title name
dc. b	'	'	;160:
dc. b	'	'	;170:
dc. b	'BR 000002-1.11'		;180:product number
			<u>;18c:version</u>
dc. w	\$f074		;18e:checksum
dc. b	'J	'	;190:i/o support
dc. l	\$00000000, \$0001ffff		;1a0:rom start, end
dc. l	\$00ff0000, \$00ffffff		;1a8:ram start, end
dc. b	'	'	;1b0:external ram information
dc. b	'	'	;1bc:modem information
blkb	\$28, ' '		;1c81ef: use prohibited
dc. b	'J	'	;1f0: corresponding contry's
			information

3) CD 2's ID Code for Japan Use

dc. b	'SEGA MEGA DRIVE	'	;100:
dc. b	'(C) SEGA 1992. DEC	'	;110: release year, month, day
dc. b	'CD2 BOOT ROM	'	;120: domestic title name
dc. b	' 12/22-1992	'	;130:
dc. b	'14:00 2.00c	'	;140:
dc. b	'CD2 BOOT ROM	'	;150: overseas title name
dc. b	'	'	;160:
dc. b	'	'	;170:
dc. b	'BR 000005- <u>2.00</u> '		;180:product number
			<u>;18c:version</u>
dc. w	\$11CC		;18e:checksum
dc. b	'JM4	'	;190:i/o support
dc. l	\$00000000, \$0001ffff		;1a0:rom start, end
dc. l	\$00ff0000, \$00ffffff		;1a8:ram start, end
dc. b	'	'	;1b0:external ram information
dc. b	'	'	;1bc:modem information
blkb	\$28, ' '		;1c81ef: use prohibited
dc. b	'J 2	'	;1f0: corresponding contry's
			information

II. Regarding DISC ID

Put the same information into I/O of DISC ID that is in MEGA DRIVE. If DISC ID is not inserted properly, you may not be able to release it. Be sure to enter the version number properly since it contains information that enables the discernment of the DISCs.

DISC ID

*	"0123456789ABCDEF"		
dc.b	"SEGA MEGA DRIVE	"	*Hardware system ID
dc.b	"(C)T-12 1992.SEP	"	*Company code, Release date
dc.b	"SEGA SAMPLE	"	*Game name for Japan
dc.b	"	"	
dc.b	"	"	
dc.b	"SEGA SAMPLE	"	*Game name for <u>Japan</u> M-CD Tech Info #3 contained an error in the left portion. <u>Overseas</u> is correct.
dc.b	"	"	
dc.b	"	"	
dc.b	"GM T - 12345 -00	"	*Disc kind, good number, <u>ver no.</u> is important.
dc.b	" <u>J6M</u>	"	* <u>I/O support</u>
dc.b	"	"	
dc.b	"	"	
dc.b	"	"	*MODEM information
dc.b	"	"	
dc.b	"	"	
dc.b	"J	"	*COUNTRY

Regarding the product ID number of the European edition's title, the following has been consolidated.

The product numbers are categorized as follows:

- | | | |
|----|-----------|-------------------------------|
| 1) | T-XXXXX | ;Third party brand |
| 2) | G-ÆÆÆÆÆ | ;SEGA brand for sale in Japan |
| 3) | MK-OOOO | ;Overseas SEGA brand America |
| 4) | MK-OOO-50 | ;Overseas SEGA brand Europe |

When dealing with the European edition's title, do not add -50 at the end of the ID product number. (By so doing, the same data would be input into the American edition.)

(Example) In the case of MK-1234-50 (European sales), the following would take place.

"GM MK - 1234 - 00 " *Disk kind, good number, ver no.

In actual terms, it would be the same if it were placed in MEGA DRIVE.

— Supplementary Remarks—

The "-00" in ID is "game's version" and the "-50" at the tail of , is "The Identification

Number for Overseas Brands of America and of Europe."

III. Regarding the Utility Within ROM

As is described in "MEGA-CD DISC 1 ¥MEGACD¥SEGA¥DOC¥README.DOC Item4," please use only the documents that our company puts out within the ROM utility. In the case where it is used after personal analysis, please acknowledge that our company cannot assume responsibility.

The routine within ROM that is presently being developed (included in MEGA-CD DISK 1 ¥MEGACD¥SEGA¥DOC¥ROMUTY) is as follows.

¥MEGACD¥SEGA¥	— — DOC — —		— — INT. DOC
			— — IO. DOC
			— — RESET. DOC
			— — SPCNT. DOC

Utility documents
within ROM.

— Caution —

Even though "MAINENT.I" is assigned, if the document within the preceding "¥ROMUTY" is not developed, it is not possible to use that utility. Further, it is possible for these documents to be added upon at a later time.

IV. Regarding the Software Manufacturing Differences of Japan's MEGA-CD/ SEGA - CD / European edition MEGA-CD

1. Development Tool

- (1) The SUPER TARGET's dip switch is changed through the hardware.

SW 4	ON: JAPAN
	OFF: USA/EUROPE

SW 5	ON: PAL
	OFF: NTSC

SW 6, 7	OFF
---------	-----

- (2) Change the BOOT-ROM above the MEGA-CD INTERFACE BOARD to BOOT-ROM that corresponds to the anticipated hardware.

For information regarding BOOT-ROM, refer to section V of this document "Types of Boot Roms."

- (3) Changing the developed system to the PAL method
Exchange the crystal chip above the SUPER TARGET to one for PAL use. For details, refer to "P 7 V. Method Change for a Developed System's NTSC PAL" of "Regarding MEGA-CD / SEGA-CD Corresponding to PAL."

2. Software

- (1) Change the security data within IP
Be cautious of the various BOOT-ROM's and security data corresponding to each country. For details, refer to "P 6 IV. Changes necessary for CD" of "Regarding the MEGA-CD / SEGA-CD Corresponding to PAL."
- (2) When changing NTSC methods to PAL methods (and back), there are things that should be done corresponding on the soft side such as the difference in the CPU clock of the hardware and the difference in the timing occurrence of V-INT. For details, refer to "Regarding MEGA-CD / SEGA - CD's Correspondence to PAL."

- (3) Differences in Corresponding Hard Reset Time

- Japanese Edition's MEGA-CD

In the early creations, when the hard's reset button was pushed, you returned to the control screen by BOOT-ROM. But because the destination can be changed when reset through application, the destination should be changed by the condition (location) where the reset button is pressed.

For details, refer to "MEGA-CD Software Creation Standard P1" and "MEGA-CD

SOFTWARE SUPPLEMENT MATERIALS."

- SEGA-CD / European Edition MEGA-CD

When the hard reset button is pressed, you return to the control screen by BOOT-ROM and the tray opens automatically. However, it is not possible to change the destination, as in the Japanese edition.

At this point, make the soft reset function wait by using the program.

As a remedy, use "In the Case where A, B, C and Start Buttons are Pressed Simultaneously" and the condition at that time (location), in order to change your destination.

For details, refer to "SEGA-CD SOFTWARE CREATION STANDARDS P2."

V. Types of BOOT-ROM

Presently, there are three types of BOOT-ROM's, distinguished by the sales region (Japan, America, Europe).

And for each country there is a merchandise BOOT-ROM and a developmental BOOT-ROM.

- Merchandise BOOT-ROM:

The BOOT-ROM is the same as the one on the merchandise's hardware. The original BOOT-ROM was directly attached to the standard board so that it could not be exchanged. However, of the hardware out in the marketplace, there are some where the BOOT-ROM portion is attached to a socket. For this instance, we have distributed these.

- Developmental BOOT-ROM

This was made for use during development and is used by setting it in the socket of the development board.

It differs from the merchandise type in the following two ways. They are identical in all other ways.

- 1) Cancel the protect on the back up RAM of the standard board for developmental use (MEGA-CD I/F board).

- If this BOOT-ROM is used, it is not necessary to input a cancel program for the protect during a game program.

- Because the protect cancel accesses address space that is not in the merchandise board hardware, do not use the developmental BOOT-ROM on the product. If it is used, there are no operations guarantee.

- 2) Do not perform the hardware's sales area CHECK.

- Originally, the hardware combination of different sales areas such as GENESIS + MEGA-CD would not start the hardware. But if this BOOT-ROM is used, even if the hardware combination includes different sales areas, it will start.

- Even in the previous instance, the software will start only if the security code of the software corresponds to the area of the BOOT-ROM that is being used.

Therefore: GENESIS
Japanese Edition MEGA-CD
Japanese Edition BOOT-ROM
Games that have US edition's security

In the case of the grouping at the above, the hardware will start but the game's software will not.

— Caution —

The BOOT-ROM here shown here is the BOOT-ROM of MEGA-CD. It does not perform the distribution of CD2's BOOT-ROM. Further, please acknowledge that the CD 2's developmental BOOT-ROM does not depend on it either.

From CS#3
CD Tech. Info #4's 1-4 2), 3) for U.S.

CD for U.S.

VER110	set 1	;0/1=Ver 1.00 / Ver 1.10
dc.b	'SEGA MEGA DRIVE '	,100
IF VER110		
dc.b	'(C)SEGA 1992.OCT '	,110 release month, year
dc.b	'SEGA-CD BOOT ROM '	,120 domestic title name
dc.b	'10/1L-1992 '	,130
dc.b	'18:30 1.10'	,140
ELSE		
dc.b	'(C)SEGA 1992.JUL '	,110 release month, year
dc.b	'SEGA-CD BOOT ROM '	,120 domestic title name
dc.b	'08/04-1992 '	,130
dc.b	'03:00 1.00 '	,140
ENDC		
dc.b	'SEGA-CD BOOT ROM '	,150 overseas title name
dc.b	' '	,160
dc.b	' '	,170
IF VER110		
dc.b	'BR 000003-1.10 '	,180 product number, version
dc.w	\$4D7A	,18e check sum
ELSE		
dc.b	'BR 000003-1.00 '	,180 product number, version
ENDC		
dc.b	'J '	,190 i/o support
dc.l	\$00000000, \$000iffff, \$00ff0000, \$00ffffff	,1a0 ROM start, end
		,RAM start, end
dc.b	' '	,1b0 outer RAM info
dc.b	' '	,1bc modem info
blkb	\$28, ' '	,1c8 - 1ff memo
dc.b	'U '	,1f0 corresponding country's info

CD2 for US

dc.b	'SEGA MEGA DRIVE '	,100
dc.b	'(C)SEGA 1993.MAR '	,110 release month, year
dc.b	'CD2 BOOT ROM '	,120 domestic title name
dc.b	'03/14-1993 '	,130
dc.b	'01:00 2.00 '	,140
dc.b	'CD2 BOOT ROM '	,150 overseas title name
dc.b	' '	,160
dc.b	' '	,170
dc.b	'BR 000006-2.00 '	,180 product number, version
dc.w	\$1e3	,18e check sum
dc.b	'JM4 '	,190 i/o support
dc.l	\$00000000, \$000iffff, \$00ff0000, \$00ffffff	,1a0 ROM start, end
		,RAM start, end
dc.b	' '	,1b0 outer RAM info
dc.b	' '	,1bc modem info
blkb	\$28, ' '	,1c8 - 1ff memo
dc.b	'U 2 '	,1f0 corresponding country's info

CD for EC

dc.b	'SEGA MEGA DRIVE'	,100	
dc.b	'(C)SEGA 1992.OCT	,110	release month, year
dc.b	'MEGA-CD BOOT ROM'	,120	domestic title name
dc.b	'10/27-1992	,130	
dc.b	'15:15 1.00	,140	
dc.b	'MEGA CD BOOT ROM'	,150	overseas title name
dc.b		,160	
dc.b		,170	
dc.b	'BR 000004-1.00	,180	product number, version
dc.w	\$ODFA	,18e	check sum
dc.b	'J	,190	i/o support
dc.l	\$00000000, \$000iffff, \$00ff0000, \$00ffffff		
		,1a0	ROM start, end
		,RAM	start, end
dc.b		,1b0	outer RAM info
dc.b		,1bc	modem info
blkb	\$28,' '	,1c8 - 1ff	memo
dc.b	'E	,1f0	corresponding country's info

CD2 for EC

dc.b	'SEGA MEGA DRIVE'	,100	
dc.b	'(C)SEGA 1993.MAR	,110	release month, year
dc.b	'CD2 BOOT ROM	,120	domestic title name
dc.b	'03/30-1993	,130	
dc.b	'12:00 2.00'	,140	
dc.b	'CD2 BOOT ROM	,150	overseas title name
dc.b		,160	
dc.b		,170	
dc.b	'BR 000007-2.00	,180	product number, version
dc.w	\$4598	,18e	check sum
dc.b	'JM4	,190	i/o support
dc.l	\$00000000, \$000iffff, \$00ff0000, \$00ffffff		
		,1a0	ROM start, end
		,RAM	start, end
dc.b		,1b0	outer RAM info
dc.b		,1bc	modem info
blkb	\$28,' '	,1c8 - 1ff	memo
dc.b	'E 2	,1f0	corresponding country's info

(END OF DOC.)

MEGA-CD / SEGA-CD SOFTWARE
PAL COVERSION

(03/10/1993)

Sega Enterprises, Ltd.

Fundamental items that are necessary for changing NTSC correspondence to PAL correspondence (and also the reverse) will be explained here.

But because games and the game system and programming methods differ, the following items may not apply to all games. There may also be various other methods.

Further, it is important to take into consideration the change from the beginning of soft development to NTSC correspondence to PAL correspondence (and also the reverse) in the development. There is the risk that changes after a game is completed can take an enormous amount of time and trouble.

I. Basic Items

	<u>NTSC</u>	<u>PAL</u>
MD Side:		
MAIN-CPU's clock	7.67 MHz	7.60 MHz
V-INT's appearance time	1/60 sec	1/50 sec
CD Side:		
SUB-CPU's clock	12.5 MHz	12.5 MHz :unchanged
INT2's appearance time (same amount of time as Main V-INT)	1/60 sec	1/50 sec
CD-DA	44.1 KHz	44.1 KHz :unchanged
PCM	12.5 MHz	12.5 MHz :unchanged

II. Points do not depend on CD. (Changing points that are the same as Cartridge/ MEGA-CD)

- Problem 1: The whole game's speed is slowed.

In the case where the game's timing depends on V-INT, the number of rotations of the programs that are executed in one second in NTSC and PAL differ. Because of this, the speed for the whole game becomes slow.

[Countermeasure (1)]

Change the speed of everything that runs at the same time as interrupt to NTSC : PAL = 1 : 1.2. For example, the transfer speed of sprite and scroll speed.

•Actual Example (1) - 1

	move.w	cnt(a6), d0	;step4
	bst.b	#06,\$a10001	;PAL(0)/NTSC(1) MAIN-
			CPU
	beq.s	?ntsc	;
?pal:	lea	tbl_pal(pc),a0	;Data TBL for PAL
	bra.s	?00	;
?ntsc:	lea	tbl_ntsc(pc),a0	;Data TBL for NTSC
?00:	move.l	(a0,d0.w),d0	;

```
tbl_ntsc:
    dc.l    $00000000
    dc.l    $00010000
    dc.l    $00020000
    dc.l    $00030000
```

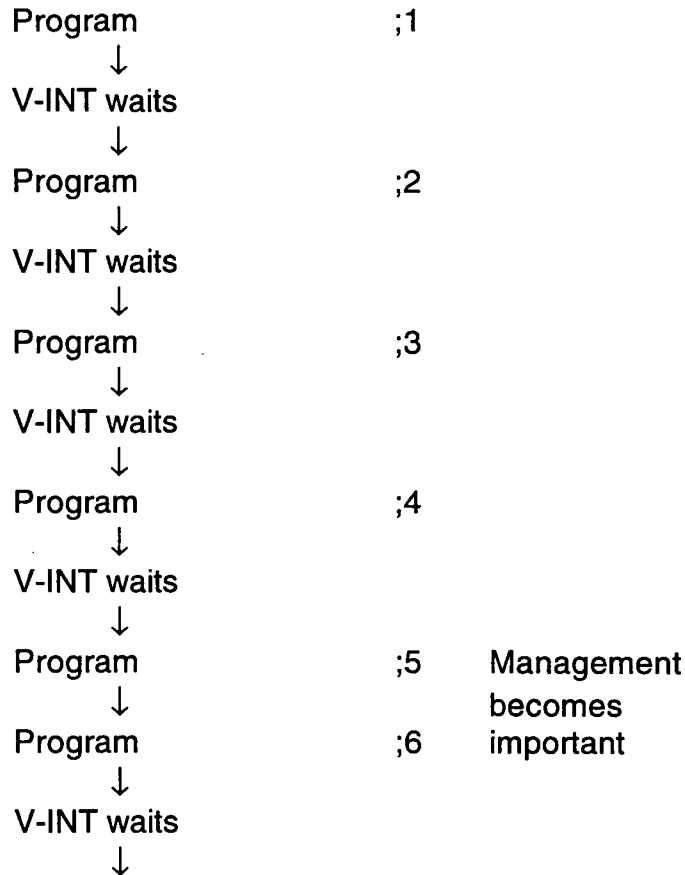
```
tbl_pal:
    dc.l    $00000000
    dc.l    $00013333
    dc.l    $00026666
    dc.l    $00039999
```

•Note 1-1

Using this method, it is not exactly the same but do not worry at the sight of it.

[Countermeasure (2)]

Only in the case of PAL, count V-INT as shown below, and during once of the five rotations, program is executed twice. (In the time that V-INT rotates five times, program is executed six times.)



Concrete Example (2) - 1

?top: *		
	*Game Program	
	btst.b #06,\$a10001	;PAL(0)/NTSC(1) MAIN-CPU
	beq.b ?ntsc	;
	addq.w #1,vint_cnt	;
	cmpi.w #6,vint_cnt	;
	bne.b ?NTSC	; Ammendment
	clr.w vint_cnt	;
	bra.b ?top	; Ammendment
?ntsc:	jsr vint-wait	;Wait V-int
	bra.s ?top	;

•Note (2) - 1

As long as you can do it during the game program, do not prohibit intrusion into V-INT.

•Note (2) - 2

In reverse, there is the possibility of using the method where PAL is drawn up as a standard, and when in NTSC, during V-INT's six rotations, the program is executed only five times.

•Problem 2: Delayed Sound

[Countermeasure (1)]

When controlling the sound on the MAIN side at 68000, draw up PAL as a standard and when in NTSC, skip one in six rotations of sound access during V-INT.

[Countermeasure (2)]

When controlling the sound on the MAIN side at Z80, NTSC and PAL's speed will not change when timed using the FM timer.

- Problem 3: The expression on the screen at 28 cell mode on a PAL monitor flickers in the bottom half (the area that is blank) of the screen when you color change during interrupt.

[Countermeasure 1]

Wait at least 3 m sec after entering interrupt (or until it actually stops flickering) and then change the color.

III. Alteration items, Cautionary items that depend on MEGA-CD

•Basic Concept

The motion timing that is the basis of the MEGA-CD side is moving at 1/75 seconds. The MEGA DRIVE side cannot decide whether this is PAL or NTSC. Further, the CD-ROM's data reproduction speed, CD-D A 's speed and PCM's reproduction speed are not influenced by PAL or NTSC. Basically, if the soft of the MAIN-CPU side is correctly managing the differences of PAL/NTSC, there is no problem. The problem occurs when there is a difference in the timings of the SUB side's basic motion timing and the MAIN side's basic motion timing, and the difference cannot be absorbed.

At this point we will use the example of the animation that is produced by the data being directly read from the CD.

•Problem 1: Regarding the number of screens of a one second period

[Countermeasure (1)]

Draw up animation that will show 10 screens in 1 second period of time.
Use WORD-RAM at 1M + 1M and perform the screen data transfer as follows.

•Concrete example (1) -1

In the methods shown in the following table, the MAIN-CPU is controlling the timing so that the screen changes constantly occurs at 1/10 seconds.

In the case of # NTSC

WORD-RAM is changed to SUB-CPU.

V intrusion 1	Screen data is transmitted to VDP
V intrusion 2	Screen data is transmitted to VDP
V intrusion 3	Screen data is transmitted to VDP
V intrusion 4	Screen data is transmitted to VDP
V intrusion 5	Wait
V intrusion 6	Return to WORD-RAM (Screen change)

Total for above table is 1/10 second.

In the case of # PAL

WORD-RAM is changed to SUB-CPU.

V intrusion 1	Screen data is transmitted to VDP
V intrusion 2	Screen data is transmitted to VDP
V intrusion 3	Screen data is transmitted to VDP
V intrusion 4	Screen data is transmitted to VDP
V intrusion 5	Return to WORD-RAM (Screen change)
Total for above table is 1/10 second.	

[Countermeasure (2)]

Perform animation by relying on the reading speed of the CD to data.

If the time it takes the MAIN-CPU side to rewrite and change one screen's worth of data is sufficiently shorter than the time it takes the SUB-CPU side to read the data, the rewriting timing will be controlled by the data reading which takes much longer. This method is performed regardless of PAL or NTSC.

Conceptual Chart

SUB-CPU		MAIN-CPU
n th number of data transferred	->	n th number of data received
n + 1 th screen reading starts		n th data screen writing starts
n + 1 th screen reading in progress		n th data screen writing in progress
.		.
.		.
n + 1 th screen reading in progress		n th data screen writing ends
n + 1 th screen reading in progress		n + 1 data screen waits, starts
.		.
.		.
n + 1 th data screen transferred	->	n + 1 th data screen received

•Note (2) -1

Even in the instance where the animation is in the n th second interval comma, each comma period's interval would not exactly equal 1/n seconds.

•Problem 2: Animation with constant sound

[Countermeasure (1)]

Leave the screen data in between the SUB side's PCM (DMA) data and the data. And when the screen's worth of data is read, pass on the data to the MAIN side. At this point, if the management time of the MAIN side is shorter than the data reading time, it is possible that it is not aware of PAL/NTSC and considers it to be a program.

However, the larger the ratio of all data that hold sound gets, the amount of data that can appear on the screen decrease. (because the amount of data that can read into a unit of time is fixed.) Because of this, the balance needs to be controlled.

IV. Changes necessary for CD

In order to become a CD-ROM for PAL use, it is necessary to change the security code and I.D. There are presently the five following types of security codes:

```
SECDB_JP DAT ;Japan use
SECDB_US DAT ;American use
SECDB_EC DAT ;European use
```

```
SECDB_JP DTU
SECDB_EC DTU
```

<*.DAT> is the smallest size. Further, <*.DTU> includes dummy data in order to match the very large-sized American type of security codes.

<Reference>

When changing the security code, one of the following of definitions of the accompanying <IP_HEAD.ASM> is necessary to make the change.

```
;US      set  1
;EUROPE  set  1
;JAPAN   set  1
```

After that, if you assemble, link, and draw up a binary file, you can quickly change the DISC image by utilizing <ISOUTIL.EXE> for the DISK image.

isoutil d:discimg,emu 1 -us ipsys.bin
will finish it. If changing the security code is all you want, it is not necessary to assemble other files and it is also not necessary to use the usual built track and built disk.

In the case where it is necessary to know whether PAL or NTSC on the SUB side, it is necessary for MAIN side to inform the SUB side so please be cautious.

V. A Developed System's Method Change From NTSC to PAL

1. Change within the same Cartridge/ CD

- (1) Substitute the crystal chip above SUPER TARGET

NTSC 53.693 MHz Õ PAL 53.2034 MHz

- (2) Change the DIP SWITCH of SUPER TARGET

DIP SW4 ON; JAPAN
 OFF; USA/EUROPE

DIP SW5 ON; PAL
 OFF; NTSC

2. In the case of MEGA-CD

- (1) Substitute BOOT-ROM above the MEGA-CD I/F board.
Three types of BOOT-ROM exist for use in Japan, US, and Europe.
(There are three types of security programs that can be entered during a game as in for ROM.)

*SUPER TARGET and MEGA-CD I/F board

When using the MEGA STATION, the above board is included in the MEGA STATION.

*The present BOOT-ROM's for overseas use and security codes are distributed only through companies that manufacture the overseas version.

*Those who wishing to obtain BOOT-ROM, security data, and crystal chips for PAL use should make their request through their company's head and contact our company's person in charge.



SEGA OF AMERICA, INC.
Consumer Products Division

MEGA CD TECHNICAL BULLETIN #9

To: Sega Developers
From: Dave Marshall, Technical Support
Date: July 6, 1993
Re: Mega CD Technical Information

=====

This document contains requirements that must be followed when producing games for the non-U.S. market.

1. All software available in Japan should not contain a "TM" or "R" with the Sega Logo. All software available for the U.S. and European markets must have a "TM" on the right side of the Sega Logo.
2. Any SEGA games made available for the European market must be PAL compatible. The PAL version must be of the same graphic and sound quality as the NTSC version.
3. Any SEGA games made available in Japan must be compatible with both NTSC and PAL game systems.