Super Gameboy notes

2010-12-27, d4s (matt@dforce3000.de)

Scope:

This document does not give a full overview of the Super Gameboy hardware and functions. Instead, it just covers the stuff that's not available elsewhere. What I found out is based on prior research by Byuu, his Bsnes sourcecode and the Super Gameboy boot ROM dumped & reverse-engineered by FPGABoy and Frode van der Meeren.

Interface:

The Super Gameboy contains a complete Gameboy sans display, buttons etc.
It is interfaced to the SNES by a chip called 1CD2-R.
There are two known versions of this chip, identifiable by their chip package designation and by reading out the version register.
The only known functional difference between them is the amount of address lines they use to decode register accesses.
I will just refer to them by their version number from now on, $21 and $61, respectively.

Registers:

The 1CD2-R doesn't use all 24 address lines to decode register accesses, which means registers are not only accessible at one specific address, but mirrored across a certain memory range:

Control registers:

-banks $00-$3f and $80-$bf
-offset $6000-$67ff

Command port:

-banks $00-$3f and $80-$bf
-offset $7000-$77ff

-When starting up, the GB CPU bootstrap code uploads 6 commands to the SGB containing data of the inserted Gameboy cartridge, range $0104-$014f(Header of Gameboy ROM and Nintendo logo).
-Each command is preceded by an identifier ($f1-$fb) and a 1 byte checksum of all values in the corresponding block added up.
-This is what the blocks look like from the perspective of the SNES CPU:

$f1	*chsum*	*[$0104]*	*[$0105]*	*[$0106]*	*[$0107]*	*[$0108]*	*[$0109]*	*[$010A]*	*[$010B]*	*[$010C]*	*[$010D]*	*[$010E]*	*[$010F]*	*[$0110]*	*[$0111]*
$f3	*chsum*	*[$0112]*	*[$0113]*	*[$0114]*	*[$0115]*	*[$0116]*	*[$0117]*	*[$0118]*	*[$0119]*	*[$011A]*	*[$011B]*	*[$011C]*	*[$011D]*	*[$011E]*	*[$011F]*
$f5	*chsum*	*[$0120]*	*[$0121]*	*[$0122]*	*[$0123]*	*[$0124]*	*[$0125]*	*[$0126]*	*[$0127]*	*[$0128]*	*[$0129]*	*[$012A]*	*[$012B]*	*[$012C]*	*[$012D]*
$f7	*chsum*	*[$012E]*	*[$012F]*	*[$0130]*	*[$0131]*	*[$0132]*	*[$0133]*	*[$0134]*	*[$0135]*	*[$0136]*	*[$0137]*	*[$0138]*	*[$0139]*	*[$013A]*	*[$013B]*
$f9	*chsum*	*[$013C]*	*[$013D]*	*[$013E]*	*[$013F]*	*[$0140]*	*[$0141]*	*[$0142]*	*[$0143]*	*[$0144]*	*[$0145]*	*[$0146]*	*[$0147]*	*[$0148]*	*[$0149]*
$fb	*chsum*	*[$014A]*	*[$014B]*	*[$014C]*	*[$014D]*	*[$014E]*	*[$014F]*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*

VRAM port:

-banks $00-$3f and $80-$bf
-offset $7800-$7fff

For each register, the following information is listed:

Name (typical access address)

access: Is register readable/writable?

ICD2-R address decoding table:

Lower address lines used by both ICD2-R versions for decoding register accesses.
Applicable values are:

-0: Address line must be low
-1: Address line must be high
-x: don't care

comment

SGB_LY_COUNTER ($6000)

access: Read

ICD2-R address decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	x	0	x	0	x
$61	0	0	0	0	x

comment:

Returns current scanline of Gameboy rendering hardware.

SGB_VRAM_UNKNOWN ($6001)

access: Write

ICD2-R address decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	1	0	0	0	x
$61	1	0	0	0	x

comment:

Exact purpose unknown, probably related to VRAM source.

SGB_CMD_READY ($6002)

access: Read

ICD2-R address decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	x	1	x	0	x
$61	0	1	0	0	x

comment:

Reads $01 if one or more commands have been completely stored in command port.
Reads $00 if all commands have been read by the SNES CPU.

SGB_CONTROL ($6003)

access: Write

ICD2-R address decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	1	1	0	0	x
$61	1	1	0	0	x

comment:

Main control register. Controls Gameboy CPU reset, Gameboy CPU clock speed, 1/2/4 player mode.

bit7:

-clearing this bit resets & halts Gameboy CPU, clears SGB_CMD_READY to $00, clears SGB_LY_COUNTER to $00.

-setting this bit starts Gameboy CPU if it was in halt state. Setting this bit after it has been set already does not reset Gameboy CPU.

-default value is clear/halted.

bit5:

-enable 4 player mode if set

bit4:

-enable 2 player mode if set

bits0,1:

-select Gameboy CPU clock divider.

-The ~21Mhz master clock of the SNES is divided by four different values according to the setting of these bits.

-*hi* designates for how many master clock cycles the GB CPU clock will stay high during each cycle.

-*lo* designates for how many master clock cycles the GB CPU clock will stay low during each cycle.

-*divider* is the number of master clock cycles one GB CPU clock cycle will take.

-*DMG CPU speed* is a rough approximation of the resulting clock speed.

bit0	***bit1***	***hi***	***lo***	***divider***	***DMG CPU speed***
0	0	2	2	/4	~5.35mhz (results in graphical glitches)
1	0	3	2	/5	~4.3mhz (normal operation)
0	1	4	3	/7	~3.05mhz
1	1	5	4	/9	~2.37mhz

SGB_JOYPAD_1 ($6004)

access: Write

ICD2-R address decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	0	0	1	0	x
$61	0	0	1	0	x

comment:

SNES writes joypad presses here.

SGB_JOYPAD_2 ($6005)

access: Write

ICD2-R address decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	1	0	1	0	x
$61	1	0	1	0	x

comment:

SNES writes joypad presses here.

SGB_JOYPAD_3 ($6006)

access: Write

ICD2-R address decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	0	1	1	0	x
$61	0	1	1	0	x

comment:

SNES writes joypad presses here.

SGB_JOYPAD_4 ($6007)

access: Write

ICD2-R adress decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	1	1	1	0	x
$61	1	1	1	0	x

comment:

SNES writes joypad presses here.

SGB_VERSION ($600f)

access: Read

ICD2-R address decoding table:

Version	***A0***	***A1***	***A2***	***A3***	***A4-A10***
$21	x	x	x	1	x
$61	1	1	1	1	x

comment:

Appears to be version register of ICD2-R. Currently known values are:

-$21: marked ***ICD2-R*** on chip package

-$61: marked ***ICD2-R*** *[stylized* ***R****(probably Ricoh logo)]* on chip package

Misc notes:
-The Super Gameboy Commander is a joypad developed and produced by Hori. Hori were also partly responsible for the design of the Super Gameboy(according to its credits roll). This joypad exploits what appears to be debugging features present in the Super Gameboy software that change the Gameboy clock speed and mute/unmute sound output.

SGB cheats (work with every known Super Gameboy software revision known so far) :

-toggle DMG speed normal/slow/very slow:

Press shoulder buttons ***l,r,r,l,l,r*** during gameplay.
This has to be entered extremely fast in order to work.
Used by Hori Super Gameboy Commander joypad.

-toggle DMG speed fast/normal/slow/very slow:

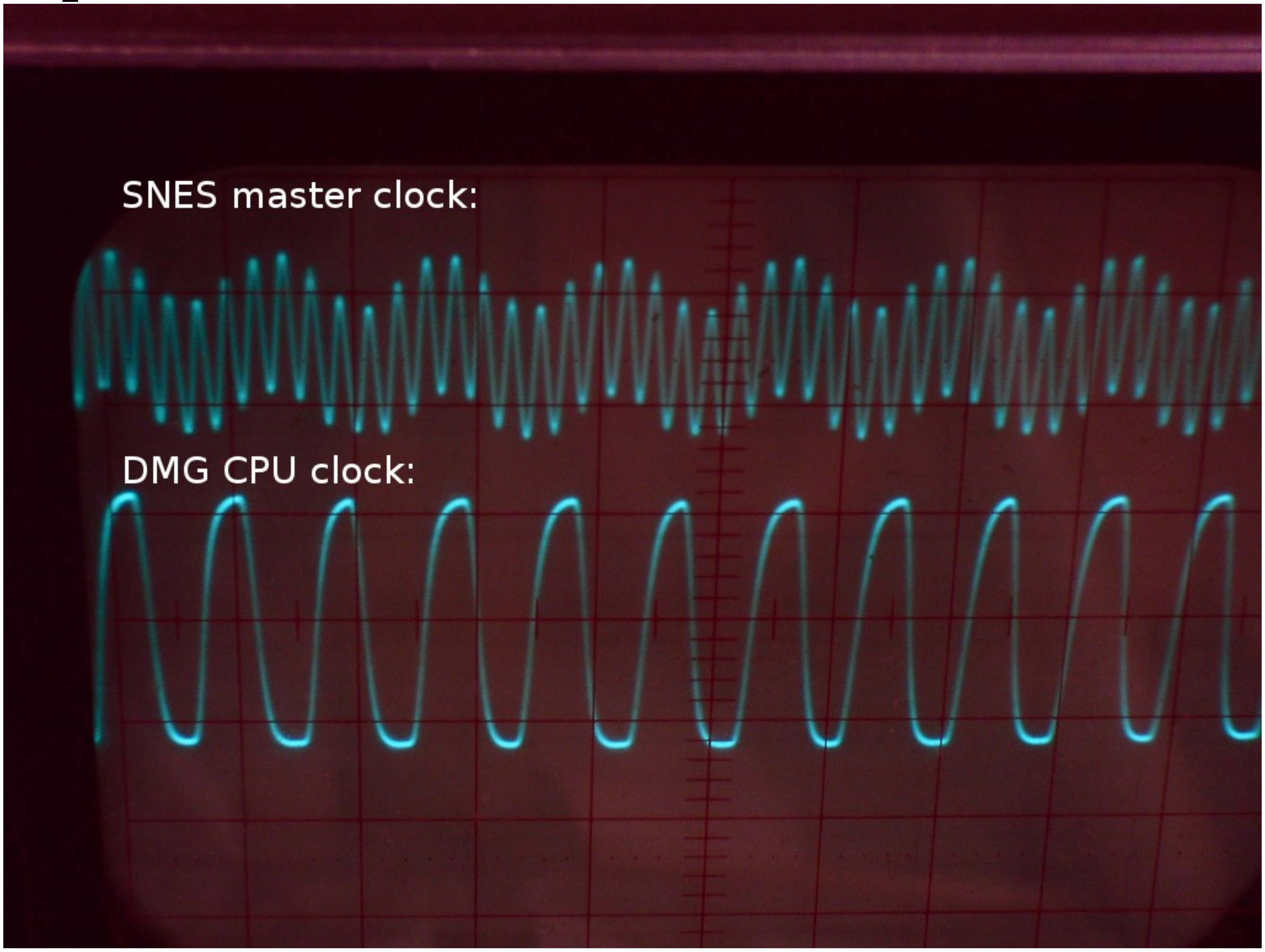
Hold directional pad up while powering on SNES,
then press shoulder buttons ***l,r,r,l,l,r*** during gameplay.
This has to be entered extremely fast in order to work.
Used by Hori Super Gameboy Commander joypad.
Fast mode prone to produce graphical glitches.

-mute/unmute SGB sound:

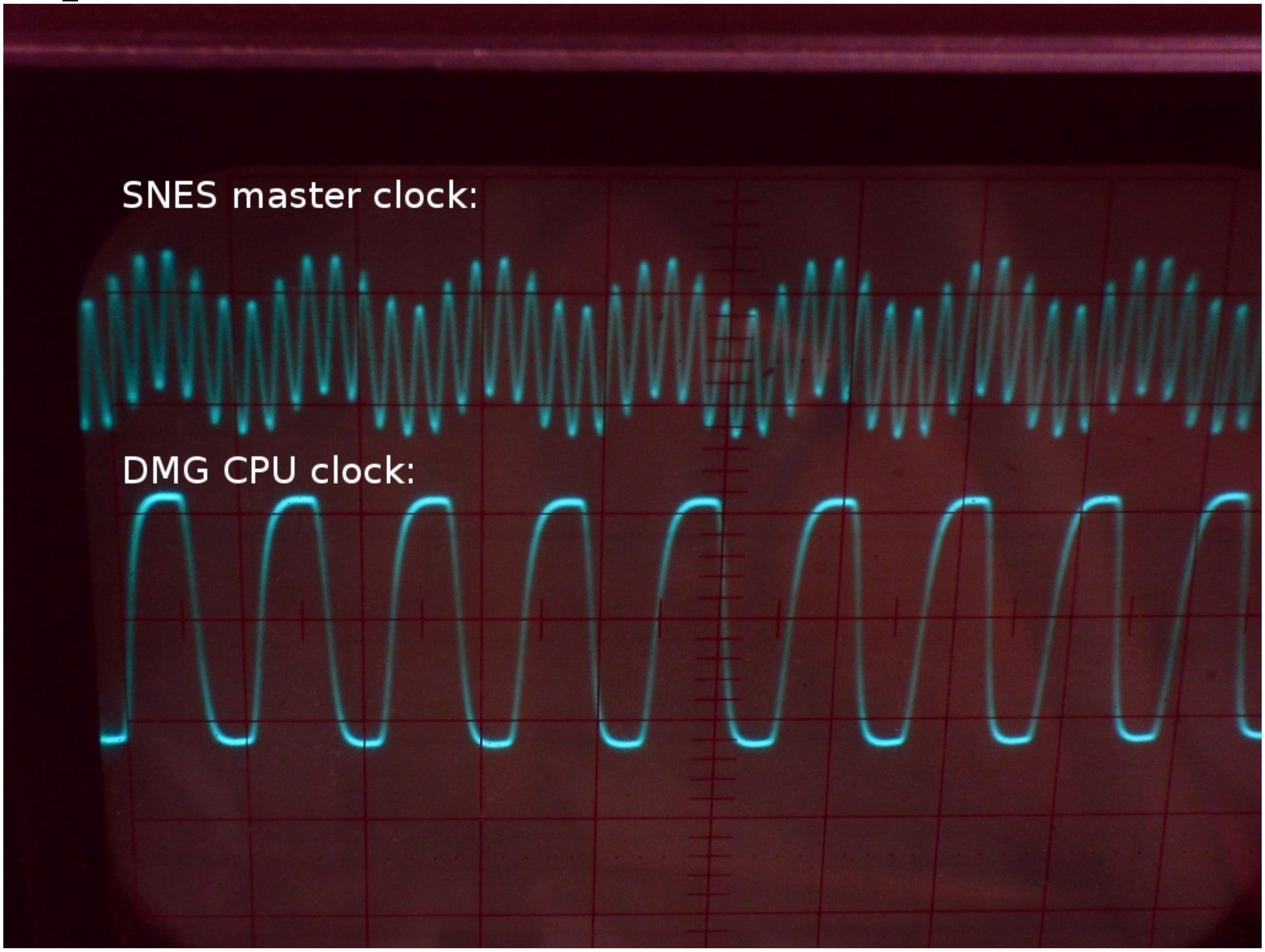
Press shoulder buttons ***r,l,l,r,r,l*** during gameplay.
This has to be entered extremely fast in order to work.
Used by Hori Super Gameboy Commander joypad.

Appendix:

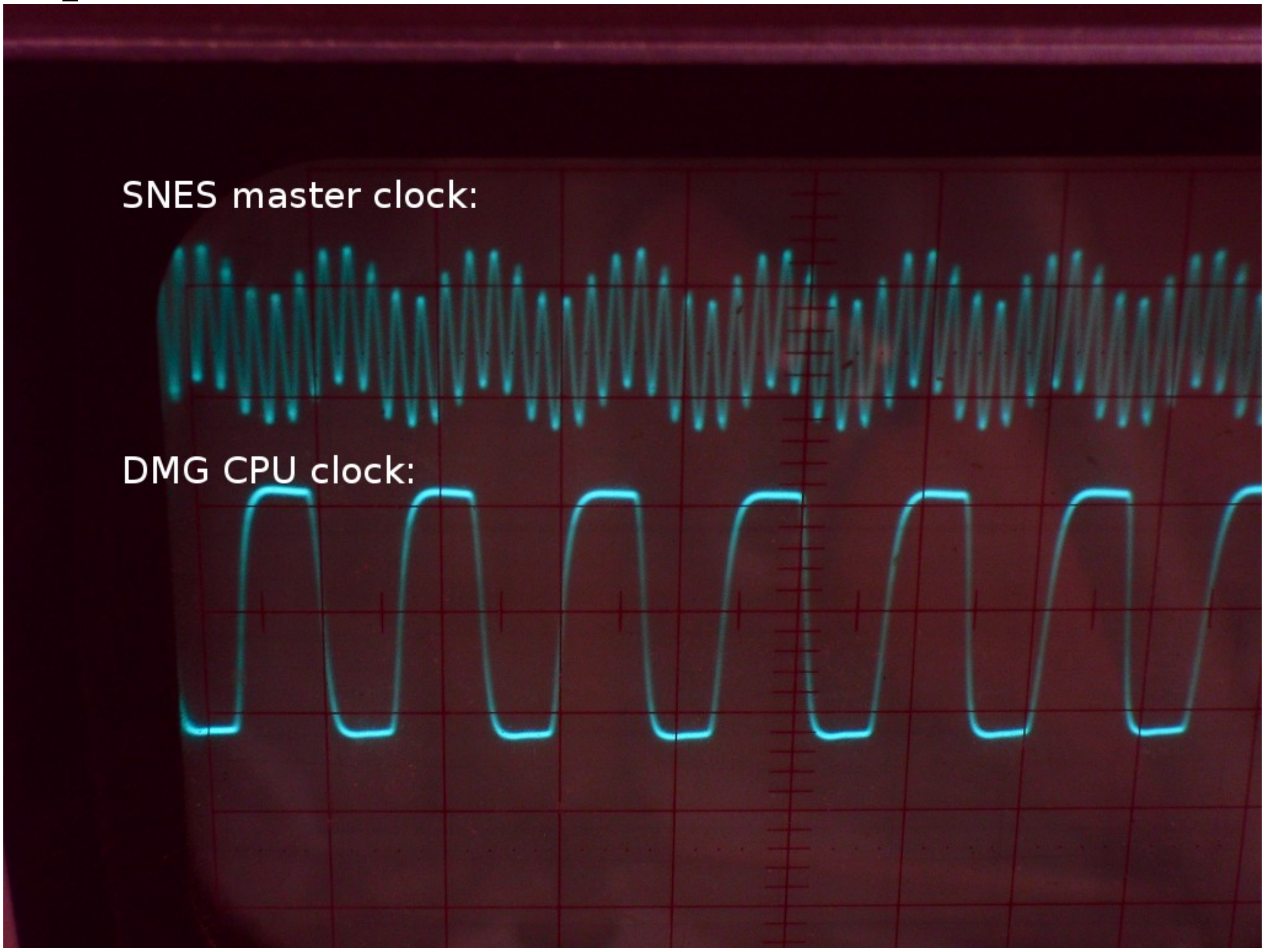
SGB_CONTROL clock divider set to $0:



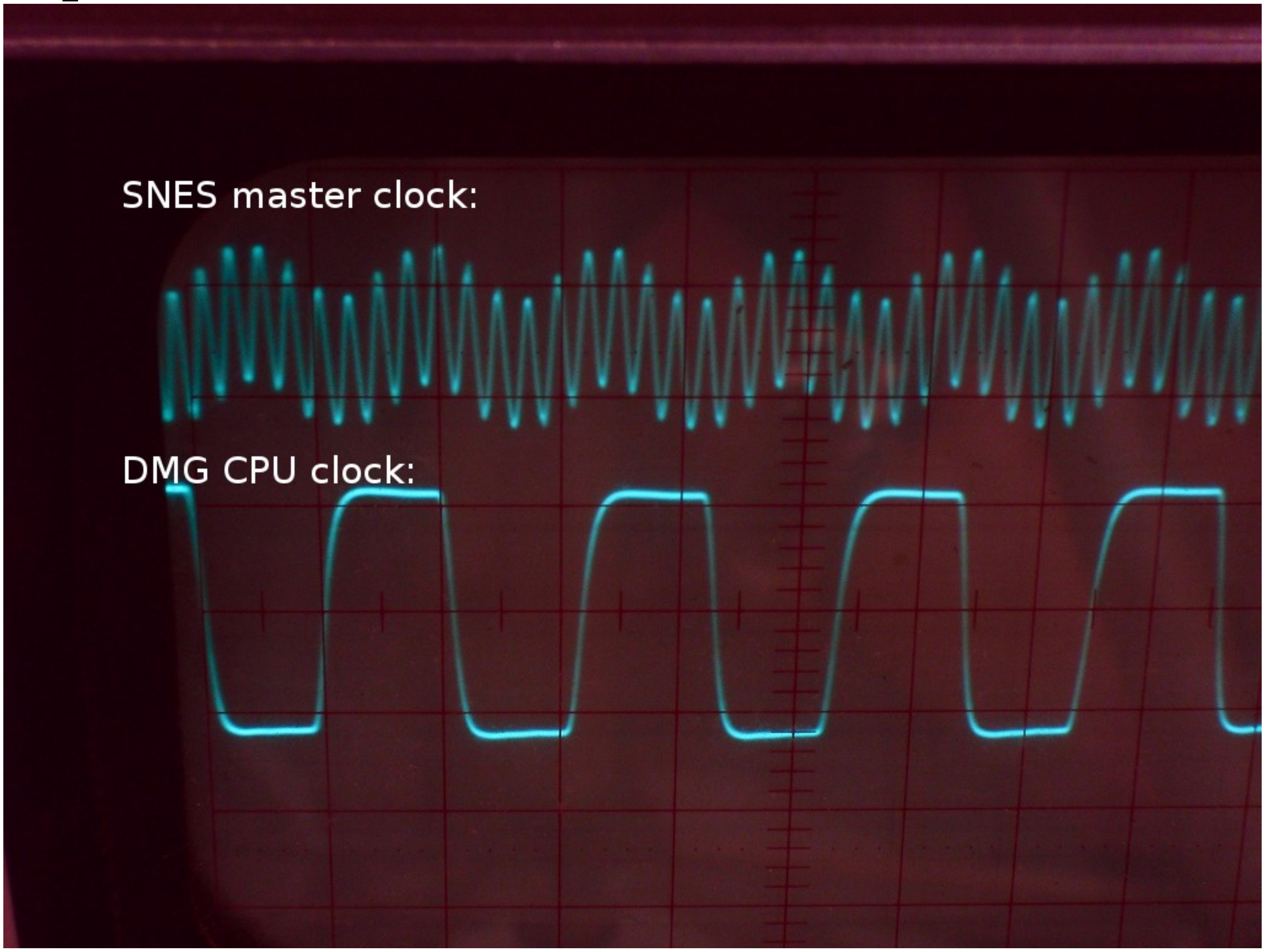
SGB_CONTROL clock divider set to $1:



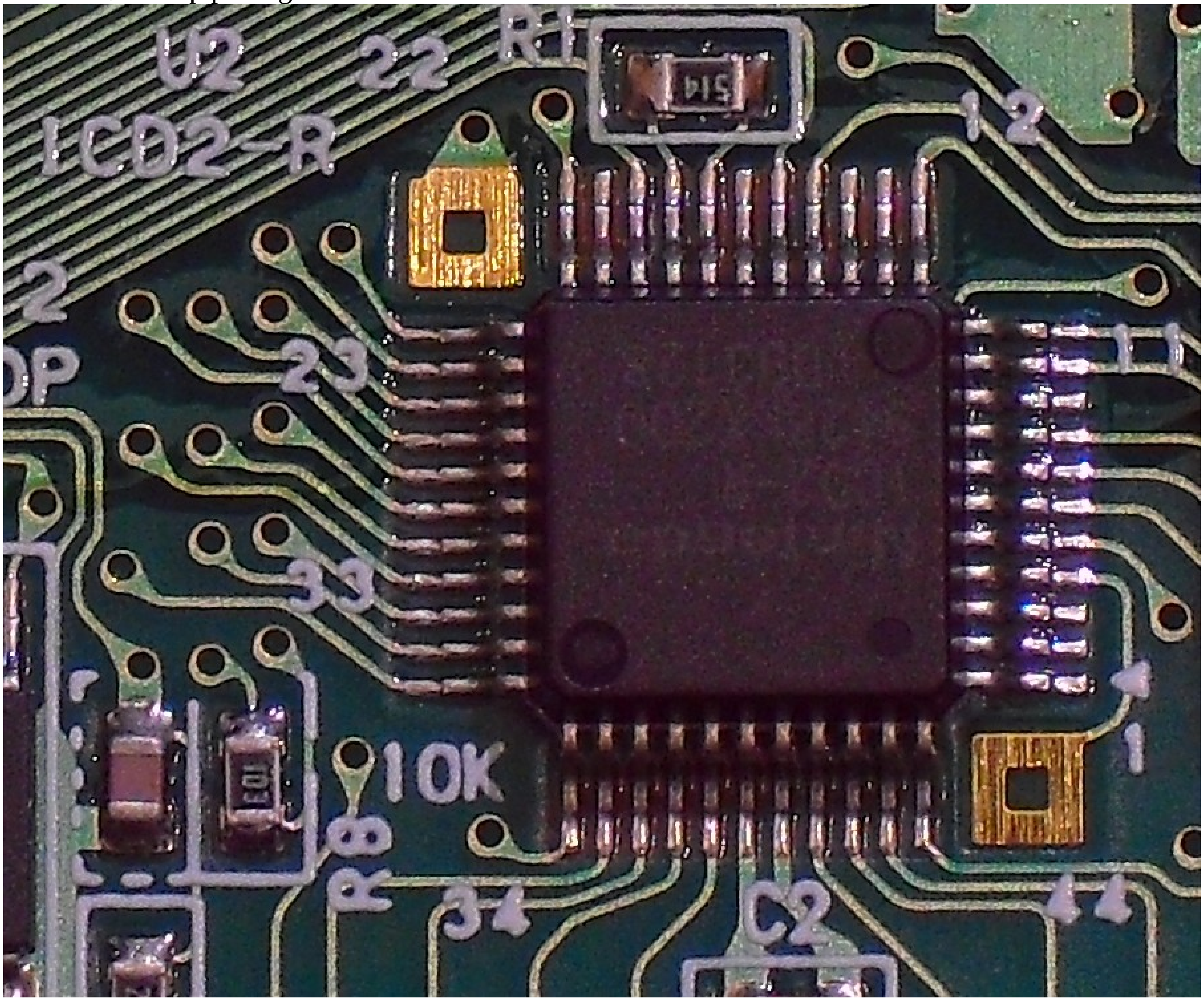
SGB_CONTROL clock divider set to $2:



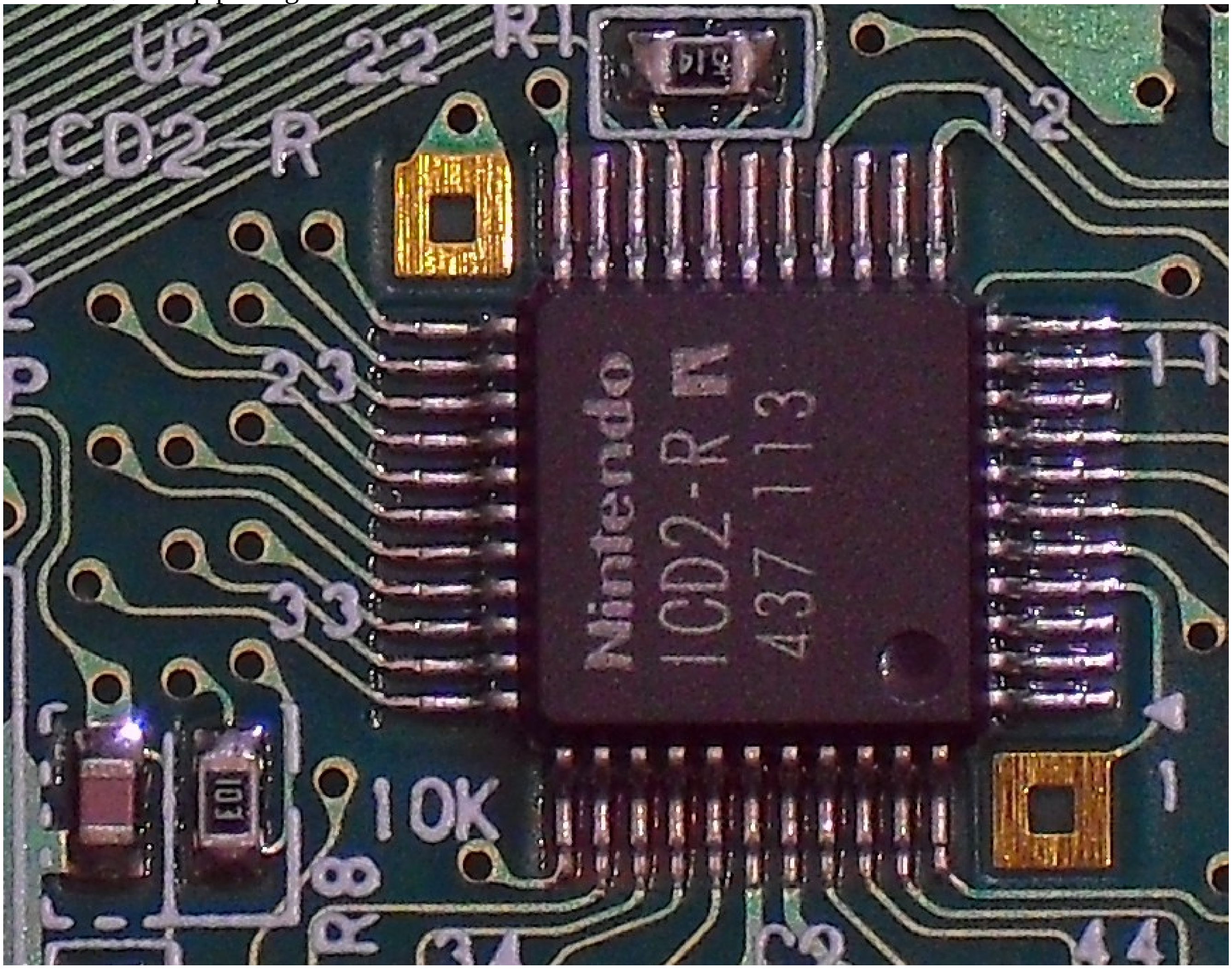
SGB_CONTROL clock divider set to $3:



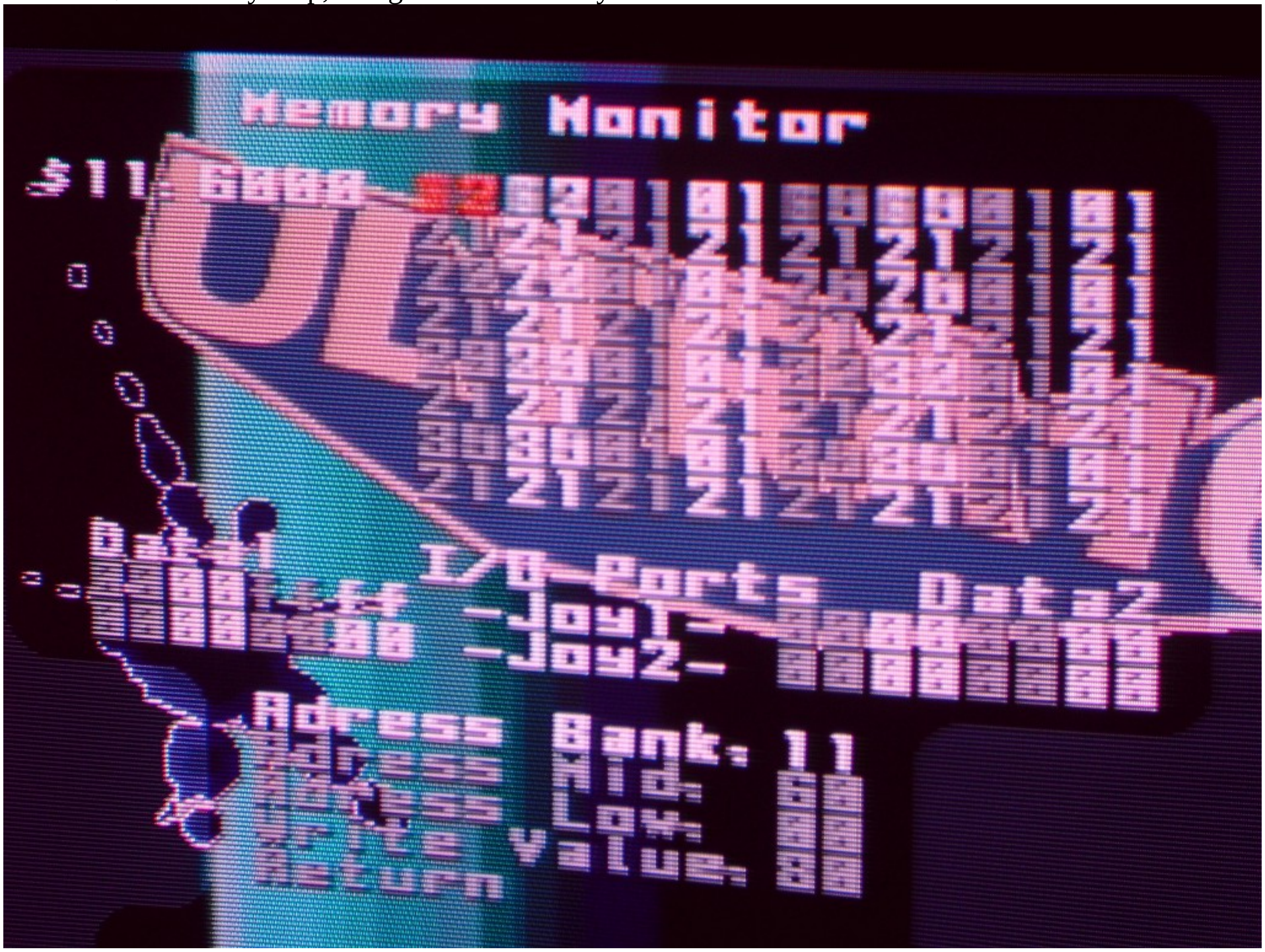
ICD2-R $21 chip package:



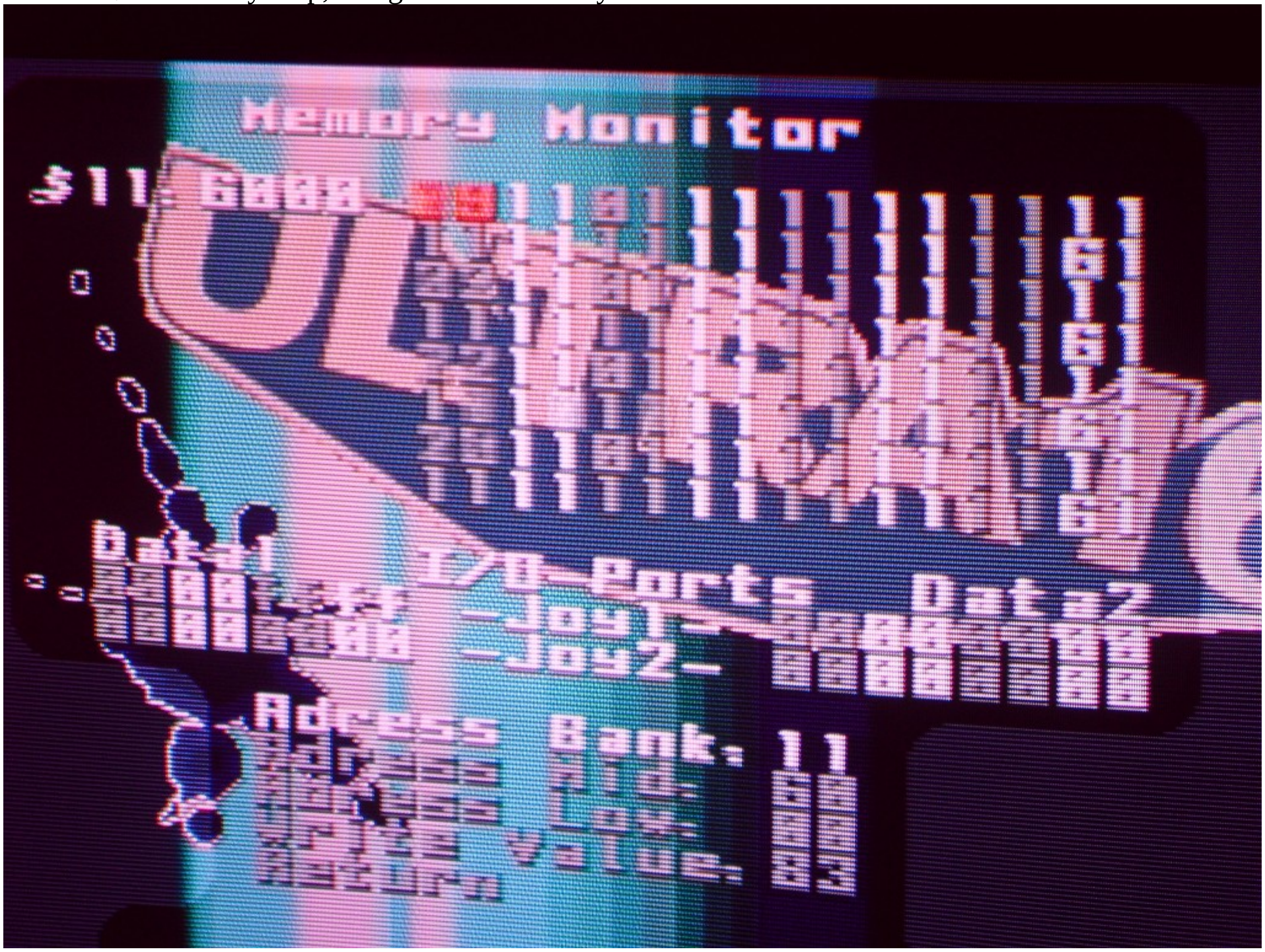
ICD2-R $61 chip package:



ICD2-R $21 memory map, using Ultra16 memory monitor:



ICD2-R $61 memory map, using Ultra16 memory monitor:



Hori Super Gameboy Commander manual(speed toggle section highlighted):

ＳＧＢコマンダー　取扱説明書

このたびはホリ電機のＳＧＢコマンダーをお買い上げいただきまして、誠にありがとうございました。
ご使用前に、必ずこの「取扱説明書」をよくお読みいただき、正しい使用法でご愛用ください。

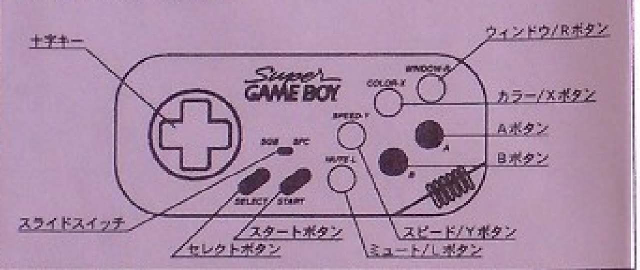
機能説明

この製品ＳＧＢコマンダーでは、スーパーゲームボーイ使用時、このコントローラーの特殊機能として下記のものがあります。

・「スピード」　ゲームスピードの切り替え
・「ミュート」　（音の　ＯＮ/ＯＦＦ）

以上の機能は、スーパーファミコンの１プレイヤー側に差しているときだけ使用できます。
２プレイヤー側では「スピード」と「ミュート」は使用することが出来ませんのでご注意ください。

各部のなまえ



スライドスイッチの説明

スーパーゲームボーイ　スーパーファミコン
SGB　SFC

●「ＳＧＢ」モードではスーパーゲームボーイ専用のボタン設定がされます。
スーパーファミコン用のソフトを「ＳＧＢ」のモードで使用されると、誤動作の原因となるのでスーパーファミコン用ソフトは「ＳＦＣ」モードでご使用ください。

☆注意事項　かならずお読みください。

スーパーゲームボーイをＳＧＢコマンダーで使うに当って以下の点に御注意ください。

・「ＳＧＢ」モードで使用するときは、スーパーゲームボーイ本体のボタン配置はＢタイプでご使用下さい。
（Ａタイプでは B ボタンが使用できなくなります）
「スーパーゲームボーイのマニュアルを読んでください。」

・スーパーファミコン用としてＳＧＢコマンダーを使用されるときは、スライドスイッチを「ＳＦＣ」でお使いください。
（「ＳＧＢ」で使用されると誤動作の原因になります）

・スーパーゲームボーイで、ＳＧＢコマンダーの機能を使用するさい、２プレイヤー側にコントローラーをセットした状態ですと「スピード」と「ミュート」の機能は使用できませんのでご了承ください。

・ＳＧＢコマンダーの機能でゲームスピードをかえられる「スピード」でゲームスピードを速くしますと、画面にノイズがでますが、故障ではありませんので安心してお使いください。

その他、スーパーゲームボーイの操作に関しては、スーパーゲームボーイの取扱説明書をお読みください。

●各ボタンの機能説明

（スライドスイッチがＳＧＢの時の説明です。）

☆WINDOW-R（ウィンドウ）

スーパーゲームボーイのウィンドウを開けたり閉めたりすることが出来ます。

☆COLOR-X（カラー）

スーパーゲームボーイで色をかえた場合の色と　もとの色（電源投入時から設定されている色）とを、入れ換えることが出来ます。

☆SPEED-Y（スピード）

＊ソフトによってはスピード切り替え出来ないものが有ります。

３段階にゲームスピードを切り替えられます。
スピード切り替えの順番は、

NORMAL → SUPER SLOW → SLOW →（NORMAL）

の順番です、「SLOW」後は「NORMAL」に戻ります。
「SUPER SLOW」は「NORMAL」よりゲームスピードが約４５％遅くなります。
「SLOW」は「NORMAL」よりゲームスピードが約３０％遅くなります。
電源投入時及び、コントローラーをスーパーファミコンに差し込んだときは「NORMAL」スピードになってます。

通常スピードは３段階切り替えですが、十字キーの上を押しながらコントローラーを差し直すか、十字キーの上を押しながらスーパーファミコンの電源を入れ直してください。スピードが４段階切り替え出来るようになります。（「DASH」モードが追加されます。）ノイズが気になるかたは、３段階モードでお楽しみください。

NORMAL → DASH → SUPER SLOW → SLOW →（NORMAL）

の順番です、「SLOW」後は「NORMAL」に戻ります。
「DASH」は「NORMAL」よりゲームスピードが約２５％速くなります。

テレビ画面ノイズ例

「DASH」のときにノイズがでますが、スーパーファミコン・テレビ・スーパーゲームボーイの故障ではありませんので、安心してお使いください。

© 1994 HORI ELECTRIC CO., LTD.

☆MUTE-L（ミュート）

音の　ＯＮ/ＯＦＦ　がボタン一つで出来ます。

（スライドスイッチがＳＦＣの時の説明です。）
通常のスーパーファミコン用コントローラーとして使用できます。ボタン配置は、
MUTE-LはＬボタン、SPEED-YはＹボタン、
COLOR-XはＸボタン、WINDOW-RはＲボタン、
SELECT・START・A・Bボタンは変わりません。

スーパーゲームボーイ・スーパーファミコンは任天堂の商標です。

定格　DC 5V　15mA

ホリ電機株式会社
本社　：〒226　横浜市緑区佐江戸町640
電話　：０４５（９３３）９６１１(代)