# open_game_module.v

## AUTHORS

### JAY CONVERTINO

## DATES

### 2025/04/19

## INFORMATION

### Brief

Colecovision Emulation of the Super Game Module using a YMZ284 for audio.

### License MIT



### open_game_module

```
module open_game_module (
input
clk,
                                                                input [
10:0]
A,
input
MREQn,
input
RFSHn,
input
IORQn,
input
WRn,
```

```
 input
 RESETn,
 input
 RDn,
                                                          inout [
 7:0]
 D,
 output
 RAM_CSn,
 output
 RAM_OEn,
 output
 AY_CSn,
 output
 AY_AS
)
```

Colecovision Super Game Module Glue Logic

### Ports

| | |
|---|---|
| **clk**<br>input | Clock for all devices in the core |
| **A**<br>input[ 10: 0] | Address input bus from Z80 |
| **MREQn**<br>input | Z80 memory request input, active low |
| **RFSHn**<br>input | Refresh request, active low |
| **IORQn**<br>input | Z80 IO request input, active low |
| **WRn**<br>input | Z80 Write to bus, active low |
| **RESETn**<br>input | Input for reset, active low |
| **RDn**<br>input | Z80 Read from bus, active low |
| **D**<br>inout[ 7: 0] | Z80 8 bit data bus, tristate IN/OUT |
| **RAM_CSn**<br>output | RAM chip select, active low |
| **RAM_OEn**<br>output | RAM Ouput enable, active low |
| **RAM_WEn** | RAM write enable, active low |
| **AY_CSn**<br>output | AY sound chip chip select |
| **AY_AS**<br>output | AY data or register select |

## REGISTER INFORMATION

Core has 3 registers at the IO addresses that follow.

| | |
|---|---|
| **SOUND_ADDR_CACHE** | h50 |
| **SOUND_CACHE** | h51 |
| **RAM_24K_ENABLE** | h53 |

**SWAP_BIOS_TO_RAM** h7F

## SOUND_ADDR_CACHE

```
localparam SOUND_ADDR_CACHE = 8'h50
```

Defines the address of r_snd_addr_cache

| SOUND ADDR CACHE REGISTER |
|---|
| 1:0 |
| CACHE LAST ADDRESS WRITE TO AY SOUND CHIP, BITS 2:1 |

Setup an address for cache the sound address so each write will be to a proper address (opcode games need to write multiple and read multiple addresses) The resister is only 4 bits since there are only 16 registers max.

## SOUND_CACHE

```
localparam SOUND_CACHE = 8'h51
```

Defines the address of r_snd_cache

| SOUND CACHE REGISTER |
|---|
| 7:0 |
| CACHE LAST WRITE TO AY SOUND CHIP |

Cache Sound Chip as the SGM games read from it (Yamaha chip does not have a read like a GI does).

## RAM_24K_ENABLE

```
localparam RAM_24K_ENABLE = 8'h53
```

Defines the address of r_24k_ena

| 24K RAM ENABLE REGISTER | |
|---|---|
| 7:1 | 0 |
| ZERO | ENABLE 24K RAM, ACTIVE HIGH |

Super Game Module 24K RAM enable using bit 0 (Active High)

## SWAP_BIOS_TO_RAM

```
localparam SWAP_BIOS_TO_RAM = 8'h7F
```

Defines the address of r_swap_ena

| SWAP BIOS TO RAM REGISTER | | | |
|---|---|---|---|
| 7:4 | 3:2 | 1 | 0 |
| ZERO | ONE | BIO TO RAM SWAP, ACTIVE LOW | ONE |

Super Game Module BIOS to RAM swap on bit 1 (Active Low)

## r_snd_addr_cache

```
reg [ 1:0] r_snd_addr_cache = 0
```

register for SOUND_ADDR_CACHE See Also: SOUND_ADDR_CACHE

## r_24k_ena

```
reg [ 7:0] r_24k_ena = 0
```

register for RAM_24K_ENABLE See Also: RAM_24K_ENABLE

## r_swap_ena

```
reg [ 7:0] r_swap_ena = 8'h0F
```

register for 8K RAM/ROM swap See Also: SWAP_BIOS_TO_RAM

## r_snd_cache

```
reg [ 7:0] r_snd_cache[3:0]
```

register for SOUND_CACHE See Also: SOUND_CACHE

# ASSIGNMENT INFORMATION

How signals are created

## s_enable

```
assign s_enable = (
RFSHn &
                                                                ~
MREQn
)
```

Decided to keep the same method used internally as the coleco. This emualtes the original ttl chip logic.

## s_y0_seln

```
assign s_y0_seln = ~(
                                                    s_enable & ~A[10] & ~
A[9] &
                                                                       ~
A[8]
)
```

Address h0000, ROM/RAM

**s_enable** Enable decoder

**A[10:8]** Address lines used for select lines (actually lines A[15:13]).

## s_ram2_csn

```
assign s_ram2_csn = ~(
                                                    s_enable & ~A[10] & ~
A[9] &
A[8]
)
```

Address h2000, RAM

**s_enable** Enable decoder

**A[10:8]** Address lines used for select lines (actually lines A[15:13]).

## s_ram1_csn

```
assign s_ram1_csn = ~(
                                                    s_enable & ~A[10] &
A[9] &
                                                                       ~
A[8]
)
```

Address h4000, RAM

**s_enable** Enable decoder

**A[10:8]** Address lines used for select lines (actually lines A[15:13]).

## s_ram0_csn

```
assign s_ram0_csn = ~(
                                                    s_enable & ~A[10] &
A[9] &
A[8]
)
```

Address h6000, RAM

**s_enable** Enable decoder

**A[10:8]** Address lines used for select lines (actually lines A[15:13]).

## s_ram_csn

```
assign s_ram_csn = (
(s_y0_seln | r_swap_ena[1]) & (s_ram2_csn | ~r_24k_ena[0]) & (s_ram1_csn |
                                                                    (
s_ram0_csn | ~r_24k_ena[0])
)
```

RAM Chip select when address is requested (active low). When the 24k is not enabled, use internal memory.

| | |
|---|---|
| **(s_y0_seln \| r_swap_ena[1])** | address range starting at h0000, swap bios/rom bit is enabled (1 is disabled). |
| **(s_ram1_csn \| ~r_24k_ena[0])** | address range starting at h4000, 24k enable bit from register. |
| **(s_ram2_csn \| ~r_24k_ena[0])** | address range starting at h2000, 24k enable bit from register. |
| **(s_ram0_csn \| ~r_24k_ena[0])** | address range starting at h6000, 24k enable bit from register. |

## RAM_OEn

```
assign RAM_OEn = RDn | s_ram_csn
```

RAM Output enable when read is requested (active low).

**RDn** Z80 read request, active low.

**s_ram_csn** See Also: s_ram_csn

## RAM_CSn

```
assign RAM_CSn = s_ram_csn
```

RAM Chip Select output assignment.

**s_ram_csn** See Also: s_ram_csn

# DECODER INFORMATION FOR SUPER GAME MODULE

How address decoder is created for Super Game Module, using a YMZ284.

**SGM IO REG** Clocked IO decoder for Super Game Module.

## AY_AS

```
assign AY_AS = (
A[7:0]
  =
= 8'h50 & ~IORQn & ~WRn ? 1'b0 : 1'b1
)
```

h50 is the address select, when selected its in data mode

**A[7:0]** If address matches h50, enable

**IORQn** Active IO request, enable

**WRn** Z80 write is active, enable

## s_ay_sound_csn

match both h50 and h51 by ignoring bit 0. Enable AY sound chip.

**A[7:0]** If address matches h50 or h51, enable

**IORQn** Active IO request, enable

**WRn** Z80 write is active, enable

## D

```
assign D = (
A[7:0]
 =
= 8'h52 & ~IORQn & ~RDn ? r_snd_cache[{2'b00, r_snd_addr_cache}] : 8'bzzzz
)
```

read cached register from previous write (AY emulation), at set address location.

**A[7:0]** If address matches h52, enable

**IORQn** Active IO request, enable

**RDn** Z80 read is active, enable