TECMO BOWL .org

Create | SBlueman

Forums | Downloads | Experiences | Online Leagues | Search...

Home Page | Member map | Online Users | Staff

Home > Forums > Hacking/Emulation > Hacking Documentation > NES RAM and MMC3 layout and registers

Mark site read

NES RAM and MMC3 layout and registers

By bruddog, January 31, 2013 in Hacking Documentation

Follow 0

Start new topic | Reply to this topic

bruddog

Down with button mashing

Moderators

3,074

11,466 posts

Location: Ca

Posted January 31, 2013

Report post

Some general infro from CXROM regarding NES ram layout and registers

I think you are confusing the cpu addressing space with the actual rom addresses. so here is a better explanation:

PC = Program Counter

PPU = Picture Processing Unit

APU = Audio Processing Unit

the NES has a 16-bit PC which holds the address of the next instruction to be executed. with a 16-bit PC you can only address $0000-$FFFF. here is a break-down of what is what in CPU address space.

$0000-$07FF: RAM

$0800-$1FFF: Mirror of RAM

$2000-$2007: PPU registers

$2008-$3FFF: Mirror of PPU registers

$4000-$4017: APU registers/sprite DMA register/controller port registers

$4018-$5FFF: for TSB, there is nothing here

$6000-$7FFF: SRAM

$8000-$9FFF: prg bank 1

$A000-$BFFF: prg bank 2

$C000-$DFFF: prg bank 3

$E000-$FFFF: prg bank 4

so the NES can only "see" $8000 bytes of data that is on the cartridge at a time (mapped to $8000-$FFFF). to get around this, they created special chips called "mappers" to swap in different banks of data into the address space the the NES can see. TSB uses the MMC3 mapper.

in my post i usually write NES addresses with a '$' (e.g. $C000) and actual file addresses with "0x" (e.g. 0x08457). now the addresses you see in a hex editor mean nothing to the NES because that data can only be mapped to $8000-$FFFF.

$4012 is an NES address. it is part of memory mapped I/O. this means that instead of having special instructions for I/O between the main cpu and the ppu/apu, the programmer just writes the data to an address. the co-processors watch for writes to those addresses and act accordingly.

now to add another layer of confusion, the MMC3 has it's own set of memory mapped registers. these registers swap PRG (code/data) and CHR (graphic) banks, changing mirroring modes (has to do with scrolling), enable/disable sram, and set IRQs (interrupt requests). these are mapped to $8000, $8001, $A000, $A001, $C000, $C001, $E000, $E001. so not only can code or data be located at these addresses, but writing to them will have some effect on the MMC3. it will not overwrite what's there since $8000-$FFFF is technically read-only.

i hope this clears up the confusion. the nesDev wiki will have a better explanation and more info if you are interested."

+ Quote

1
kamphuna8 reacted to this

Reply to this topic...

GO TO TOPIC LISTING

RECENTLY BROWSING 1 MEMBER

SBlueman

Home › Forums › Hacking/Emulation › Hacking Documentation ›
NES RAM and MMC3 layout and registers

✓ Mark site read

Theme ▾ Contact Us