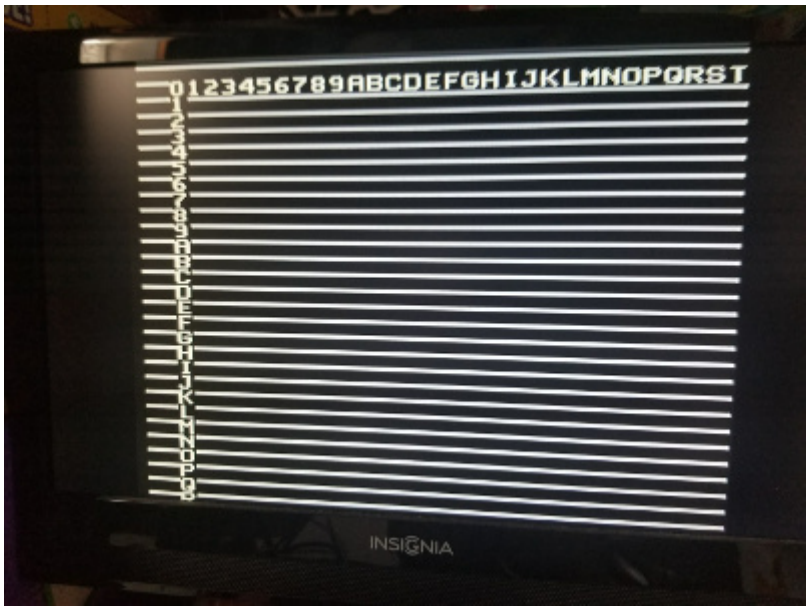
nesdoug

PPU writes during rendering.

I'm going to do some experiments today. Writing to the PPU during rendering.

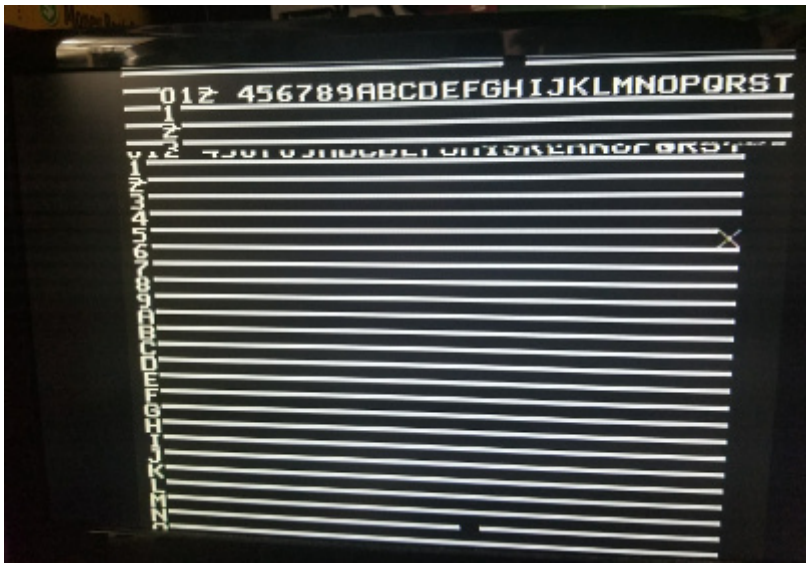
Normally, you never want to do this. The NES was designed so that you write to the PPU during v-blank or when rendering is off. Let's see what happens, with the ultimate goal of changing 1 color mid-screen without ruining the picture.

Here's a simple asm6 template code.



With a timed loop that waits till we are definitely not in v-blank. I'm trying to write an X near the top left. Emulators have been disappointing on these tests, so I have to go right to actual hardware (US NTSC top loader to a flat screen TV).

```
lda #$20
sta $2006
lda #$42
sta $2006
lda #4 ;is an X
sta $2007
```



What's happening here, is writing twice to 2006 during rendering has overwritten the scroll registers and misaligned the screen. Also, our X is nowhere near the top left. The last point, is where emulators failed accuracy…some of them DID put the X on the top left.

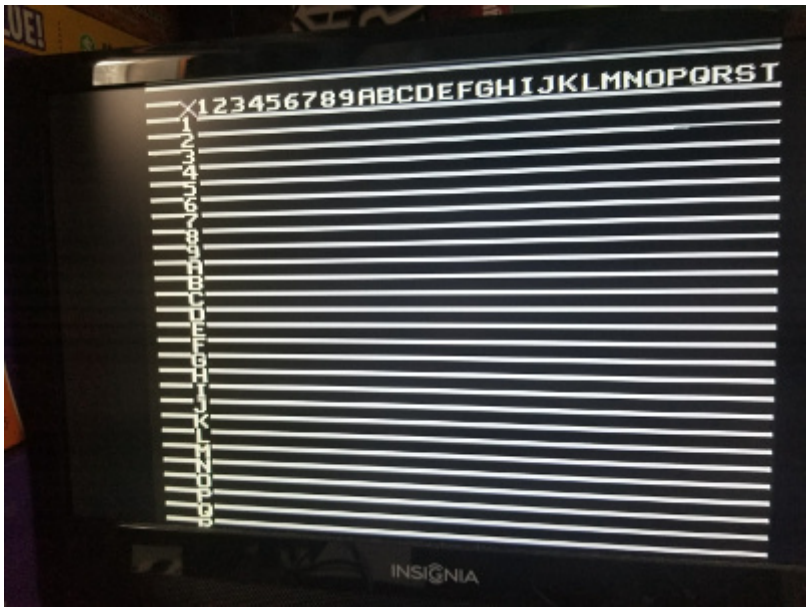
So, let's try to fix that. After writing the X, we will then write a new 2006 2006 to try to realign the screen…and time the write near the far right of the screen to reduce visible glitches.

```
lda #$20
sta $2006
lda #$42
sta $2006
lda #4 ;is an X
sta $2007

lda #$20
sta $2006
lda #$a0
sta $2006
```

The write is still not going to the right place, nor is it an X. And the 3 is squished at the top, as it is skipping a scanline. Let's try turning rendering off…doing the write…turn rendering on…fix scroll with 2 2006 writes. I also had to move the scanline wait a little.

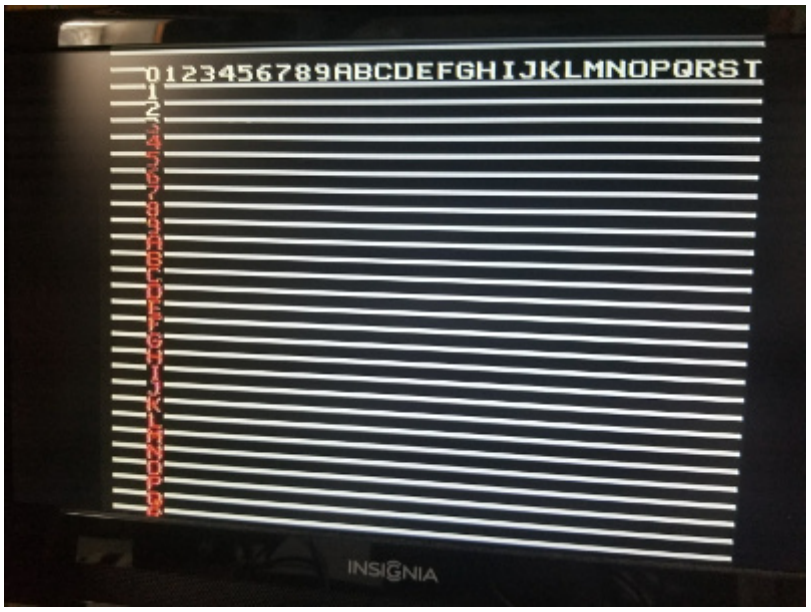
```
lda #$20
sta $2006
lda #0
sta $2001 ;rendering off
lda #$42
sta $2006
lda #4 ;is an X
sta $2007
lda #$1e
sta $2001 ;rendering on
lda #$20
sta $2006
lda #$a0
sta $2006
```



Ok, that seems to work, except with some noticeable glitches. Perhaps better timing could reduce that as well. I'm not too concerned about that right now…so on to using the same setup to change a color. I want all numbers below the 2 to be red.

First I have to write $30 (white) to the text palette during NMI, so the top stays white…then rendering off…change color…rendering on…fix scroll.

```
lda #$3f
sta $2006
lda #0
sta $2001 ;rendering off
lda #$03
sta $2006
lda #$16 ;red
sta $2007
lda #$1e
sta $2001 ;rendering on
lda #$20
sta $2006
lda #$a0
sta $2006
```



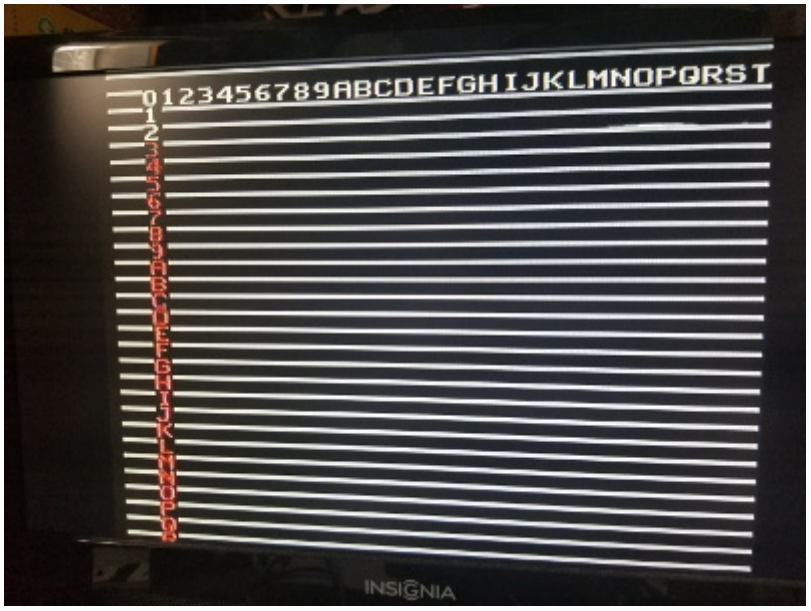
Not perfect. The top of the 3 is still white.

Moving the split up is not enough, since that will just cut the top of the 3 off. We must do a fancy 2006 2005 2005 2006 write, about 2 scanlines earlier. See the nesdev wiki on x/y scroll splits…

https://wiki.nesdev.com/w/index.php/PPU_scrolling#Split_X.2FY_scroll (https://wiki.nesdev.com/w/index.php/PPU_scrolling#Split_X.2FY_scroll)

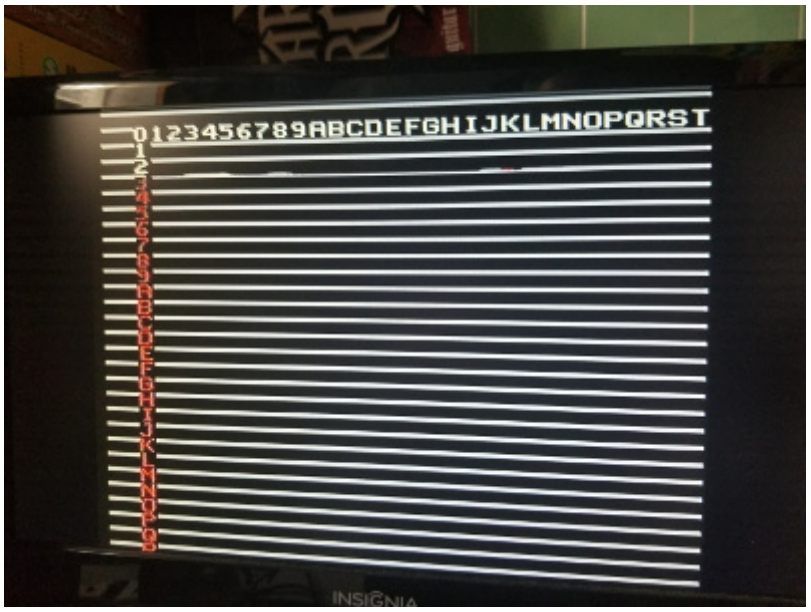
```
lda #$3f
sta $2006
lda #0
sta $2001 ;rendering off
lda #$03
sta $2006
lda #$16 ;red
sta $2007
lda #$1e
sta $2001 ;rendering on

lda #0
sta $2006
lda #40
sta $2005
lda #0
sta $2005
lda #$a0
sta $2006
```



That seems to be working. There are still some noticeable glitches though. Probably, this could be improved. Maybe another day I will try to hide it. But even still, timing THIS fragile is dangerous. This is all done with timed loops. What if something interrupts us?

I added a DMC sample to fire every second. The DMC channel, when ON, steals cycles from the CPU to read a byte…which ruins perfectly times loops.



So, if you used a MMC3 scanline counter, and could time this to perfection…you could make mid-screen changes beyond just scroll shifts. But, it could be error prone. Many of my tests ended with the PPU writes going to the wrong address. Rendering needs to be off, meaning you should only do this on a blank part of the screen.

Here's the source code. In 6502 asm for the asm6 assembler. Caution, I just wrote / rewrote some of this code (especially the controller read code) and it hasn't been thoroughly tested.

download (http://dl.dropboxusercontent.com/s/hfhrazfvvspzrja/PPUwrites.zip)

Furthermore…I forgot that turning rendering off makes sprites behave erratically. This sort of thing may only be possible on screen with no sprites. I will have to test this some more.

March 21, 2018March 21, 2018 dougfraker