
The Skinny On NES Scrolling

Documento original: The Skinny On NES Scrolling
Autor: Loopy
Traducido por: <Hyde> (líder del proyecto *Translatore*).
E-Mail: Hyde@programmer.net
Página en la WEB: http://hydesprojects.cjb.net/

Versión: 1.0
-Primera versión de esta traducción.

Parte I

La información disponible sobre el *scrolling* del *background* es suficiente para la mayoría de los juegos. Sin embargo, este documento explicará otro mecanismo utilizado para ejecutar esta tarea (de *scrolling*).

Estos son los registros relacionados con la misma:

- (V) Dirección de la VRAM, la cual es determinada por el registro \$2006 (16 bits)
- (T) Dirección temporaria de la VRAM (16 bits) (Uno puede decir que esta dirección esta compuesta por 15 bits, ya que el último bit no es usado).
- (X) Desplazamiento X (horizontal) en un *tile*.

La PPU usa la dirección que se encuentra en el registro \$2006 para leer y escribir datos en la VRAM y para sacar bytes de *name tables* que son usados para dibujar el *background*. La PPU actualiza la dirección (V) usada para la extracción de los datos de *name tables* mientras ella misma dibuja el *background*. Los bits 0-11 de (V) contienen la dirección del *name table* que esta siendo usado (-\$2000). Los bits 12-14 contienen el desplazamiento Y (vertical) en el *tile*.

Acciones que afectan los contenidos de los registros:
(Aquí se usara notación lógica, ya que esto simplifica todas las explicaciones).

(D) **Byte** escrito/leído.

- \$2000 (**Escribiendo**):
(T):0000110000000000 = (D):00000011
- \$2005 (**Escribiendo por primera vez**):
(T):0000000000011111 = (D):11111000
(X) = (D):00000111
- \$2005 (**Escribiendo por segunda vez**):
(T):0000001111100000 = (D):11111000
(T):0111000000000000 = (D):00000111
- \$2006 (**Escribiendo por primera vez**):
(T):0011111100000000 = (D):00111111
(T):1100000000000000 = 0
- \$2006 (**Escribiendo por segunda vez**):

(T):0000000011111111 = (D):11111111

(V) = (T)

Principio de un scanline (suponiendo que el background y los sprites están activados):

(V):0000010000011111 = (T):0000010000011111

Principio de un frame (línea 0) (suponiendo que el background y los sprites están activados):

(V) = (T)

Parte II

Uno puede interpretar los bits 0-4 de la dirección de la VRAM como el desplazamiento $X * 8$ (*X scroll*), el cual es incrementado por la PPU mientras los gráficos son dibujados. Cuando este número cambia de 31 a 0 el bit 10 de esta misma dirección es permutado. Esto explica como cambios horizontales de un *name table* a otro (#1 a #2 y #2 a #3) ocurren.

Uno puede interpretar los bits 5-9 de la dirección de la VRAM como el desplazamiento $Y * 8$ (*Y scroll*). Este *scroll* funciona de una manera diferente. El bit 11 de la dirección es permutado cuando este número es incrementado desde 29 a 0 (en vez de 31 a 0). Esta acción tiene ciertos efectos colaterales: si uno deja este valor ser mayor que 29 (a través de \$2005 o de \$2006), el cambio de 29 a 0 no ocurrirá y datos de un *attribute table* serán usados como datos de *name table*. El desplazamiento (Y) aun cambia de 31 a 0, pero sin permutar el bit 11. Esto explica el porque de cuando uno recibe un valor de desplazamiento (*scroll*) negativo al escribir un número mayor que 240 en 'Y' en el registro \$2005.

Observaciones:

-\$2005 y \$2006 comparten el "permutador" que determina la cantidad de veces que uno ha escrito algo en estos registros. Leyendo los contenidos del \$2002 hará con que este "permutador" sea "limpiado" (de ON para OFF, o de OFF para OFF).

Ilustración:

Permutador: OFF

\$2005 (Escribiendo)

Permutador: ON.

\$2005 (Escribiendo): Segundo byte es escrito ya que el permutador estaba en ON.

Permutador: OFF.

\$2005 (Escribiendo)

Permutador: ON.

\$2006 (Escribiendo): Segundo byte es escrito en \$2006 ya que el permutador estaba en ON.

Permutador: OFF.

\$2005 (Escribiendo)

Permutador: ON.

\$2002 (Leyendo)

Permutador: OFF.

\$2005 (Escribiendo) Primer byte es escrito en \$2005.

Permutador: ON.

-Los desplazamientos (X e Y) pueden ser interpretados de la siguiente manera:

<--X-->

```
00011000  ^
00011000  |
00011000  |
00011000  |
00011000  Y
00011000  |
00011000  |
00011000  |
```

Supongamos que estos bytes forman un *tile*. Ahora supongamos que el valor dentro de (**X**) es 1. Entonces el *tile* sería dibujado de la siguiente manera:

<--X-->

```
00001100  ^
00001100  |
00001100  |
00001100  |
00001100  Y
00001100  |
00001100  |
00001100  |
```

La interpretación del desplazamiento Y es análoga a esta.

Terminología:

Background- Pantalla de fondo.

Sprite- Estas son las figuras que están en movimiento constante en los juegos (Mario es un *sprite* en el juego "Super Mario Brothers").

Tile- 8x8 píxeles. Todos los gráficos de los juegos están almacenados en *tiles*.

Scroll- Desplazamiento en la pantalla de fondo.

Name table- Tablas que contienen bytes de información sobre la pantalla de fondo.

Attribute table- Lea el NESTech para obtener mas información.

¿No te gustó lo que leíste? ¡Envíame tus sugerencias!