

Blitter Test Files

This page contains information about the Williams Special Chips, which were 'bit blitters'- block transfer chips that could move data around on the screen and in memory faster than the CPU. In fact, I've timed the special chips at 16 megs in 18.1 seconds. That's 910K/sec, not bad for the early 80s.

The blitters were not used in Defender and Stargate, but were added to the ROM boards of the later games. Splat!, Blaster, Mystic Marathon and Joust 2 used Special Chip 2s. The only difference that I've seen is that SC1s have a small bug- D2 is inverted. So when you tell the SC1 the size of the data to move, you have to exclusive-or the width and height with 4. The SC2s eliminate this bug.

The blitters were accessed at memory location \$CA00-CA07. CA01 is the mask, usually \$FF to move all bits. CA02-3 is the source data location. CA04-5 is the destination data location. CA06 is the width, and CA07 is the height. Writing to CA00 starts the blit, and the byte written determines how the data is blitted.

Bit 0 indicates that the source data is either laid out linear, one pixel after the last, or in screen format, where there are 256 bytes from one pair of pixels to the next. Bit 1 indicates the same, but for the destination data.

I'm not completely sure what bit 2 does. This test didn't show any difference with it on or off, but the source was always in ROM. While working on my new multigame software I tried some blits from RAM. If bit 2 was on, they worked fine. If it was off, garbage was blitted. Maybe it changes timing?

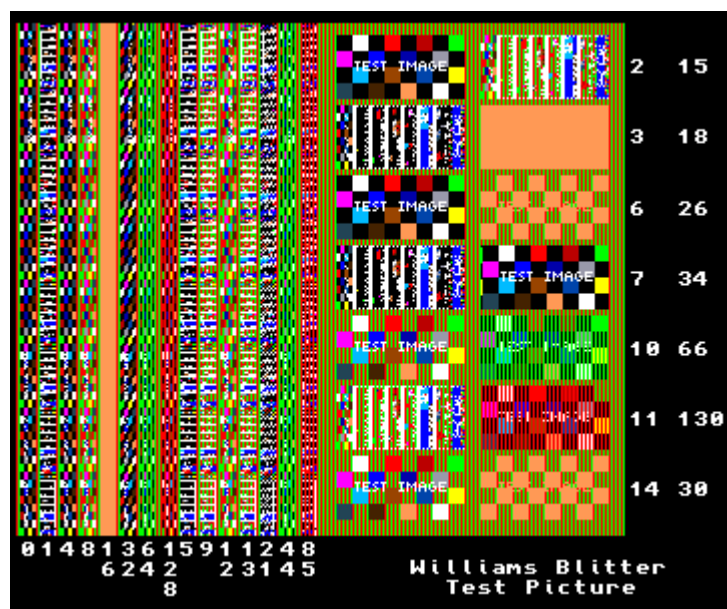
Bit 3 tells the blitter only to blit the foreground- that is, everything that is not color 0. Also known as transparency mode.

Bit 4 is 'solid' mode. Only the color indicated by the mask is blitted. Note that this just creates a rectangle unless bit 3 is also set, in which case it blits the image, but in a solid color.

Bit 5 shifts the image one pixel to the right. Any data on the far right jumps to the far left.

Bits 6 and 7 only blit every other pixel of the image. Bit 6 says even only, while bit 7 says odd only.

Here is a picture from the RAM dump that I created. The program blitted a 64x32 pixel image 29 times over a background of alternating red and green pixels. The numbers below and to the right of the picture show what value was written to \$CA00 to initiate the blit.



[My program](#)

[The 6809DASM .dat file](#)

[The RAM dump](#)

[back to Williams Info page](#)

[back to Arcade Game Info page](#)

[back to Home page](#)