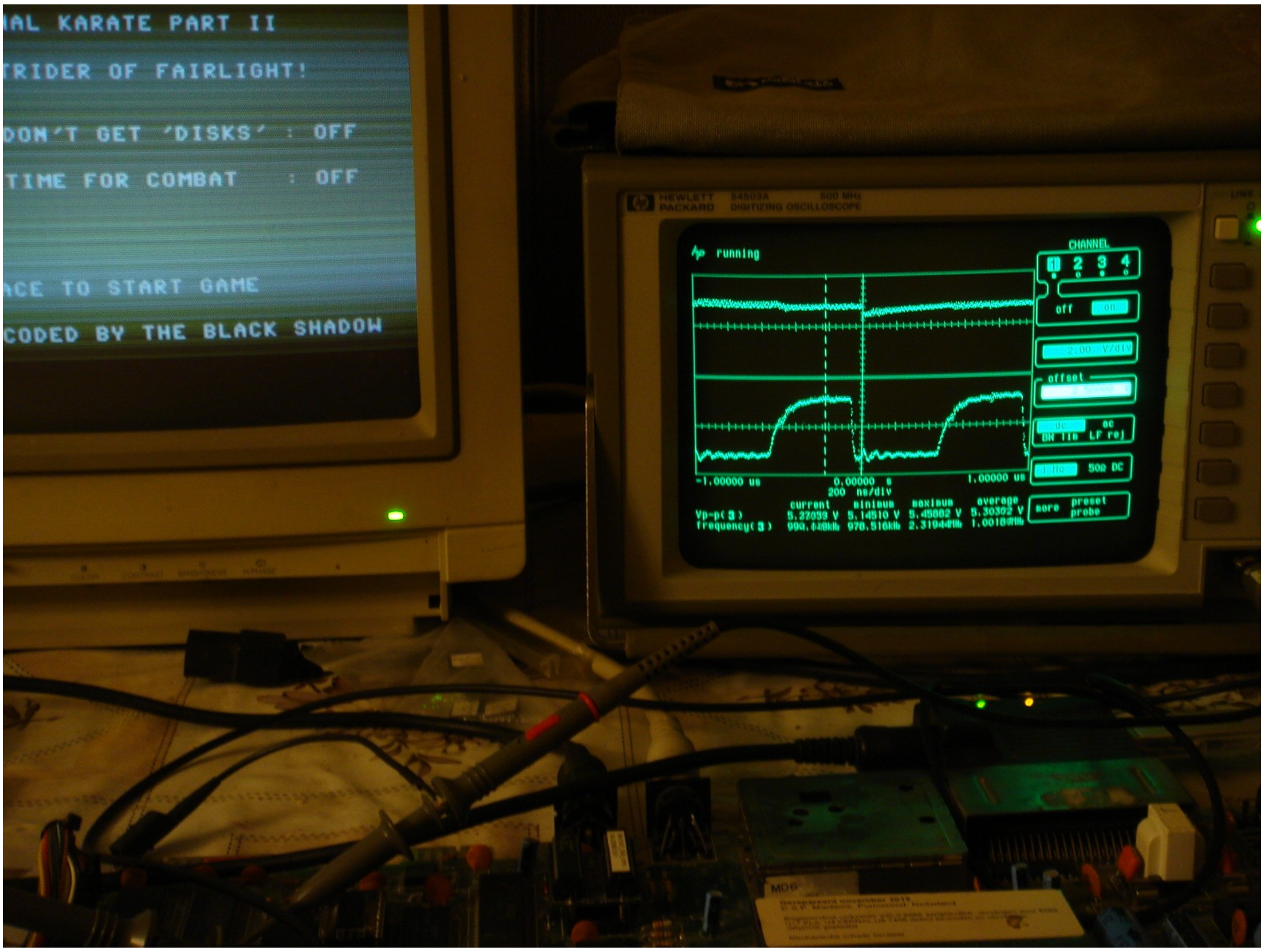
The “Ghostbusters” issue

Some users of the PLA20V8 have reported crashes in Ghostbusters. These issues always went away after trying different GAL chips or switching between the “base” and “alt2”. I do not own any hardware where I can trigger the Ghostbusters issue. The issue has been rare enough to not consider it a design problem in the PLA20V8, but nevertheless it has remained a mystery.

The “I/O glitch”

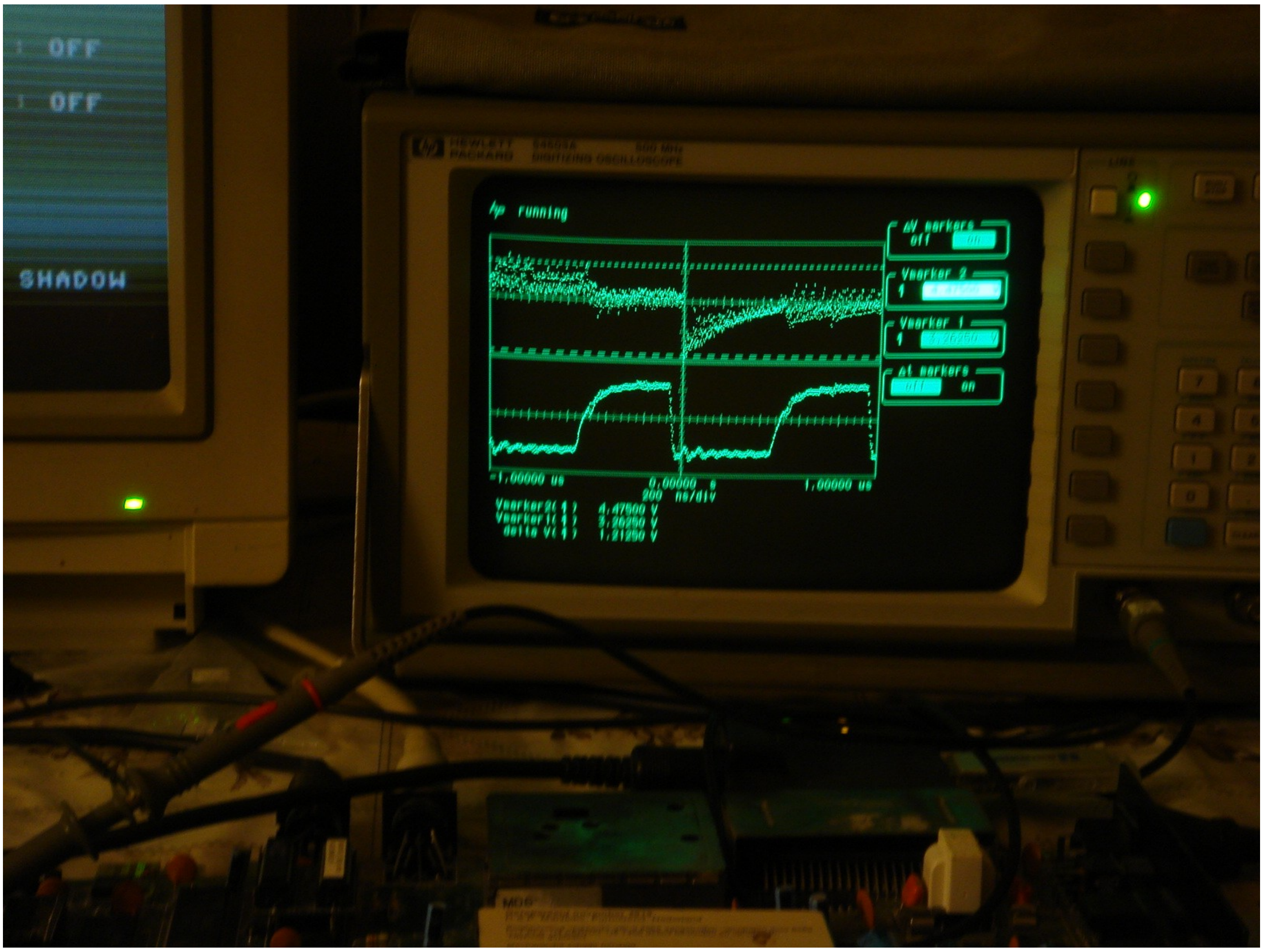
When measuring the chip select of the I/O chips of the Commodore 64, sometimes a small trench appears on the oscilloscope. For example I am starting the Fairlight release of International Karate + (https://csdb.dk/release/?id=22533) and when on the trainer selection screen I am measuring the CS line of the SID chip on the top channel and PHI2 on the bottom channel of my oscilloscope:



No music is playing here, so the SID chip should not get selected. In the above example I was using an Assy 250407 with a PLA20V8 with 25ns GALs and the “base” file installed. The oscilloscope was set to trigger on the upper channel at a trigger level of 3.75V.

I call this phenomenon the “I/O glitch”. The I/O glitch is not something specific to the PLA20V8, the original Commodore 906114 shows exactly the same behaviour.

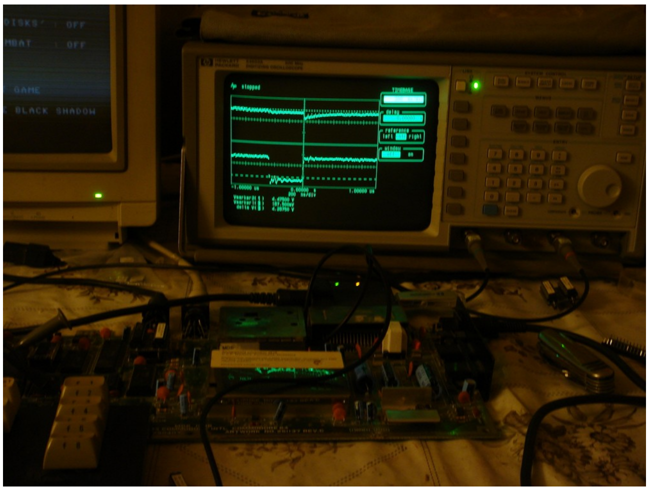
Let’s zoom in on the I/O glitch:



I have placed a marker line at the deepest point of the trench. This is line at 3.26 volt. 3.26 volt is well above the minimum “TTL high” level of 2.0 volt, therefore the trench is harmless and will not trigger any chip.

The cause of the I/O glitch

In the following photo I am showing the I/O glitch again on the SID CS line at the top channel, and at the bottom channel the I/O output of the PLA. This reveals the cause of the I/O glitch:

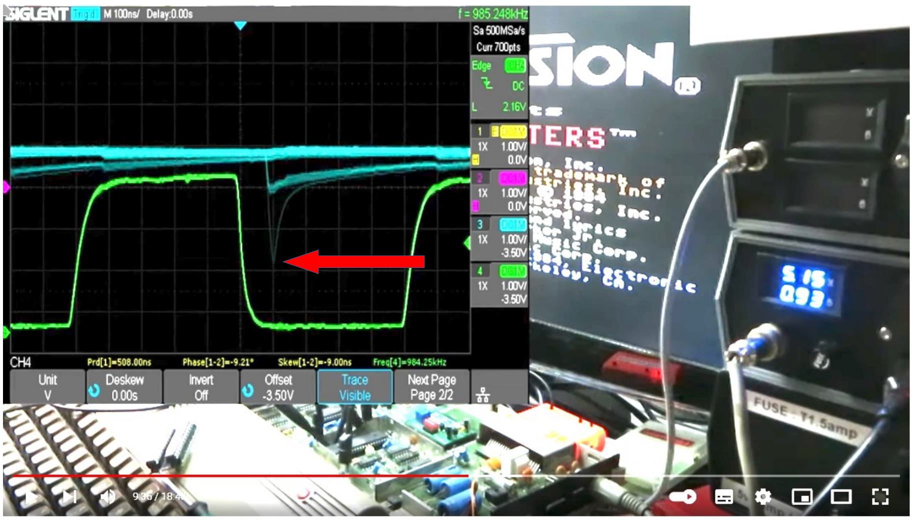


The I/O line was active and this activation ends when the VIC-II uses the AEC signal to take control of the bus. The CPU responds to the chip by releasing the bus and the PLA responds to this by deselecting I/O line.

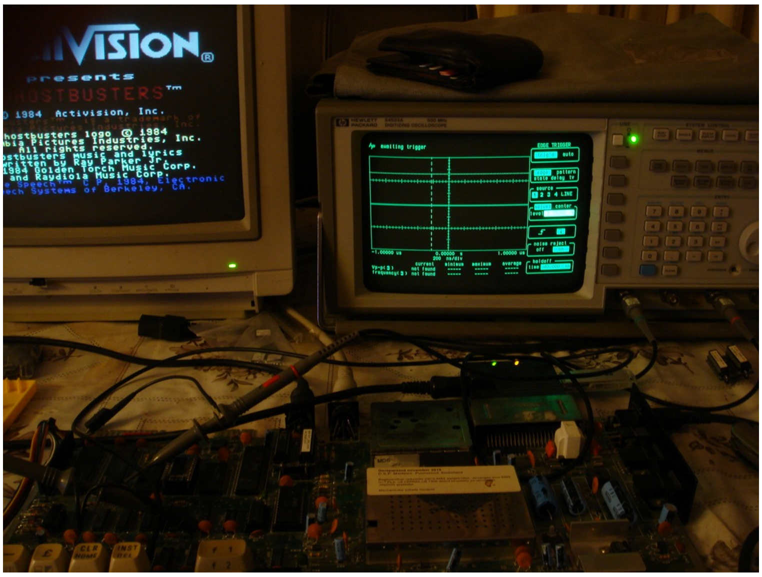
The 74LS139 on the C64 mainboard is has an enable pin which is connected to the PLA I/O line. This means that when the I/O line becomes deselected, the 74LS139 disables any CS signal to an I/O chip. However, when AEC happens, a lot happens on the bus, i.e. the address bus is transitioning. Transitioning address lines just before the I/O line gets deselected are the cause of the I/O glitch.

The I/O glitch on Lee Ashmore’s machine

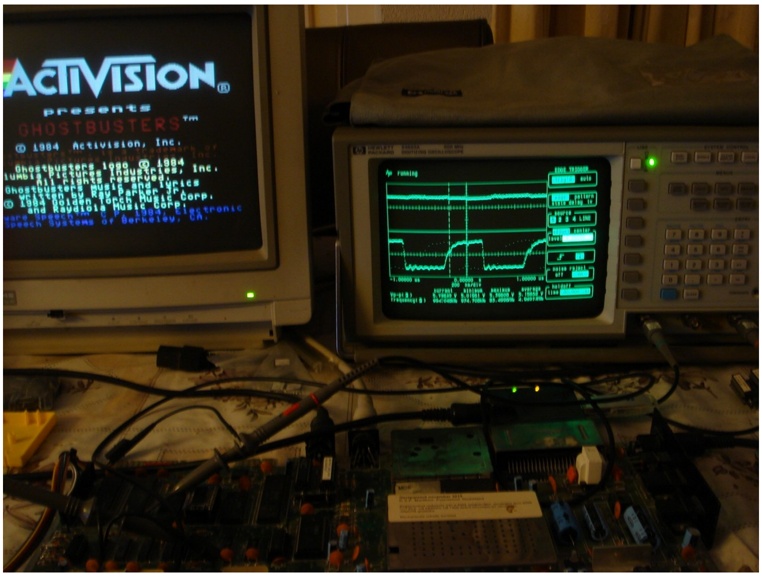
Lee Ashmore’s C64 is sensitive to the Ghostbusters issue. Very interesting is that his machine is showing a much deeper I/O glitch trench. This was measured with the original Commodore PLA:



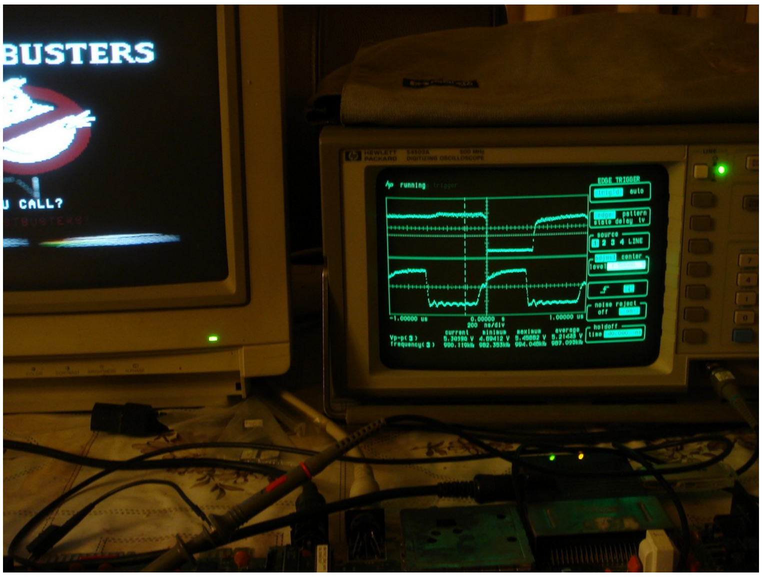
His machine was showing the intro screen of Ghostbusters. I don’t know what the deepest voltage of this trench is, but it is clear it won’t need to go a lot deeper in order to trigger a TTL low. So if the PLA20V8 just goes slightly deeper (I didn’t see it happen in the video), it actually could trigger a chip. This deeper trench is already interesting, but what is even more remarkable is that on my machine, on the same screen, I am unable to see any I/O glitch at all:



The trigger level of the oscilloscope is set to 3.75V and it keeps waiting indefinitely for a trigger. Only when increasing to 4V it does trigger, but then just shows the stable high logic level:



For information, this is how it looks when the game has started and the music is playing (trigger at the normal 1.5V for TTL signals):

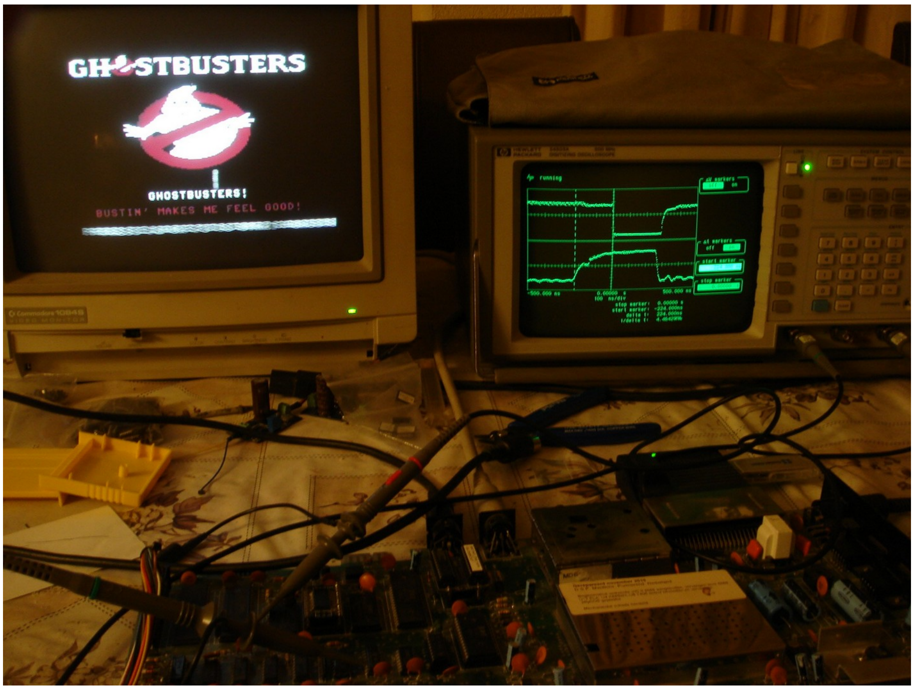


So we have a machine with much deeper I/O glitch trenches and the Ghostbusters issue together. Could both issues be related?

Fixing the I/O glitch

Because the I/O glitch exists in the original Commodore 64 PLA and such glitches can be considered undesired, the I/O glitch can be considered a flaw in the logic of the Commodore 64 PLA. It is not difficult to fix the I/O glitch. By using the CAS signal to clock the I/O chip select, it is possible to prevent that the I/O line is active when the AEC signal switches.

The following photo is taken with a PLA20V8 with such a fix. The result is that the SID I/O line is activated slightly later during the cycle:



And this means that I am no longer able to get any I/O glitch in situations where I got them before: