OpenCore Probleme auf uralter Hardware

Beitrag von „Hecatomb“ vom 27. Mai 2022, 19:17

ich hab was gelesen das der usbinjectall Kext da nicht gebraucht wird. welches usb hat er denn verbaut?

- FakePCIID_XHCIMux.kext

`8086:1e31` `8086:9c31` `8086:9cb1` `8086:9c31` `8086:8cb1`

This injector is a bit of an extension to normal FakePCIID duties. It doesn't actually fake any PCI IDs. Rather, it forces certain values to XUSB2PR (PCI config offset 0xD0) on the Intel XHCI USB3 controller. The effect is to route any USB2 devices attached to the USB2 pins on the XHC ports to EHC1. In other words, handle USB2 devices with the USB2 drivers instead of the USB3 drivers (AppleUSBEHCI vs. AppleUSBXHCI).

So normally what is a complex "multiplex" DSDT patch (that is not well understood), is a simple kext install.

Configuration properties and their defaults:

RM,pr2-force <00 00 00 00>. By default forces all XHCI ports to route USB2 devices to EHC1.

RM,pr2-init <01>. Will write RM,pr2-force value at startup if non-zero.

RM,pr2-block <01>. Will block writes to XUSB2PR if non-zero.

RM,pr2m-block <01>. No evidence that OS X drivers attempt to write XUSB2PRM (offset 0xD4), but since this kext relies on a valid value here (as provided by the BIOS), writes to it are blocked if non-zero.

RM,pr2-honor-pr2m <01>: Changes to XUSB2PR will be masked by XUSB2PRM if this is non-zero.

RM,pr2-chipset-mask: Writes to XUSB2PR are masked by this value. This is defined by the chipset documentation. Default value depends on chipset.

Refer to Intel 7/8/9-series chipset data sheet for more info.

In order to create your own injector, you should be familiar with IOKit matching and kext Info.plist files. There is ample documentation available on developer.apple.com. Use the existing injectors as a template to build your own.

kommt das so hin?

oder vielleicht das? USB-Fix für El Capitan V1