

Erledigt

Probleme mit GenericUSBXHCI.kext und USB Audio Interface

Beitrag von „luki13“ vom 10. Dezember 2014, 15:59

Hallo,

Ich habe mir kürzlich einen 'MacPro' mit einer Haswell-E CPU und X99 Chipsatz zusammengestellt (steht alles in der Signatur). Mavericks läuft unter Clover auch sehr schnell und stabil.

Nun habe ich jedoch bei längerem Arbeiten das Problem, dass das externe Audiointerface, das über USB angesteuert wird, nach einiger Zeit keinen Ton mehr abspielt, bzw. nur noch Bruchstücke.

Nach langem Probieren fand ich raus, dass es an der GenericUSBXHCI.kext liegt. Ohne diese läuft alles ohne Ausfälle. Jedoch bringt diese kext dem System einiges an Schub. Mit geladener kext erreicht der Benchmark 20000, ohne jedoch nur 15000.

Mit chimera tritt der Audioausfall schon nach wenigen Sekunden auf. Auch hier entscheidet die kext wesentlich über die Benchmarkergebnisse.

Habt ihr vielleicht Ideen oder wisst was da schief läuft?

Danke für die Hilfe.

Mit freundlichen Grüßen

luki13

Beitrag von „neo_io“ vom 4. Februar 2015, 23:58

Hi luki13,

ich habe sehr wahrscheinlich das selbe Problem wie du. Nach oft nur wenigen Sekunden bekomme ich immer heftigere Verzerrungen im Sound und dann ist irgendwann nichts mehr zu hören.

Ich hatte erst auch die Vermutung, dass es mit der GenericUSBXHCI.kext zusammenhängt.

Hast du schon eine Lösung gefunden, ich habe das Problem sowohl mit als auch ohne der GenericUSBXHCI.kext.

Ich benutze ein Focusrite Scarlett 6i6 USB Audio Interface mit der neuesten Beta Version 1.9.64.4 der MixControl Software (Version 1.8 habe ich den selben Fehler)

edit:

hatte einen Fehler beim deinstallieren der .kext! Nach der Deinstallation läuft es scheinbar auch bei mir störungsfrei. Muss das aber morgen nochmal in Ruhe testen. Wie hast du die bzw. welche Performance getestet?

Danke! MfG neeo_io

Beitrag von „luki13“ vom 7. Februar 2015, 23:20

Hallo neeo_io,

ich habe jetzt herausgefunden, dass es bei mir, auch wenn die GenericUSBXHCI.kext nicht installiert ist, es ebenfalls nach einiger Zeit zu Problemen kommt. Es verzögert lediglich den Eintritt. Hat es bei dir geklappt? Ich habe mit und ohne die .kext einen Geekbench Benchmark gemacht und hatte eine Differenz von 5000 Punkten, was schon gravierend ist. Wie sieht es bei dir aus?

Beitrag von „Griven“ vom 7. Februar 2015, 23:32

Ähm...

Geekbench misst die Prozessor und Ram Leistung und hat mal so ganz und gar nichts mit USB zu schaffen sprich ob nun der GenericUSBXHCI.kext installiert ist oder nicht hat nichts mit der Geekbench Performance zu tun es sei denn am USB3 hängt noch eine bisher unentdeckte CPU...

Beitrag von „luki13“ vom 7. Februar 2015, 23:38

Deshalb wundert es mich ja 😊

Ich habe sonst nichts am System verändert.

Kext geladen: 20000

Kext nicht geladen: 15000

Kann das vielleicht mit dem relativ neuen und nicht bekannten Prozessor zusammenhängen?

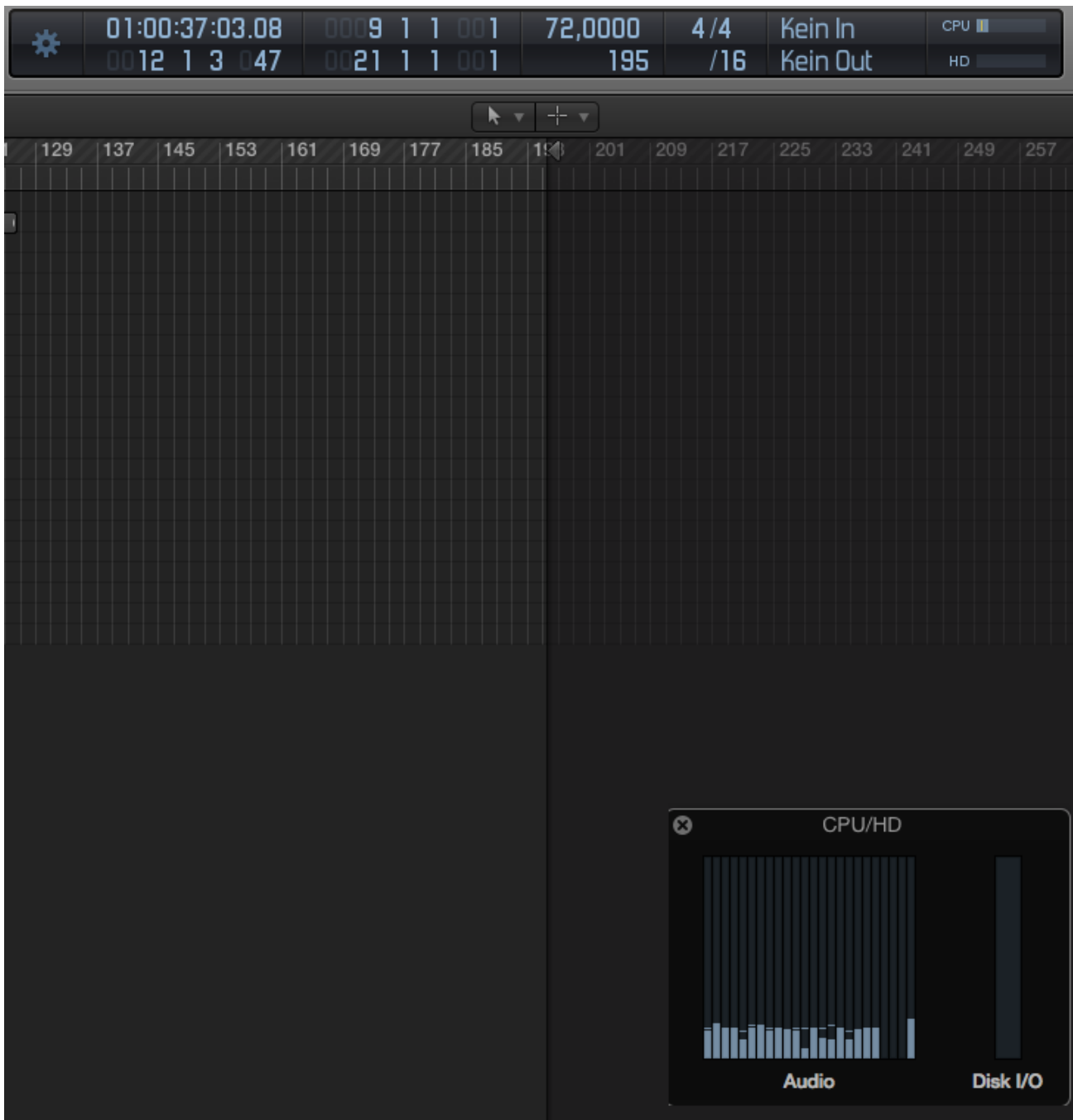
Beitrag von „DSM2“ vom 15. Mai 2016, 11:35

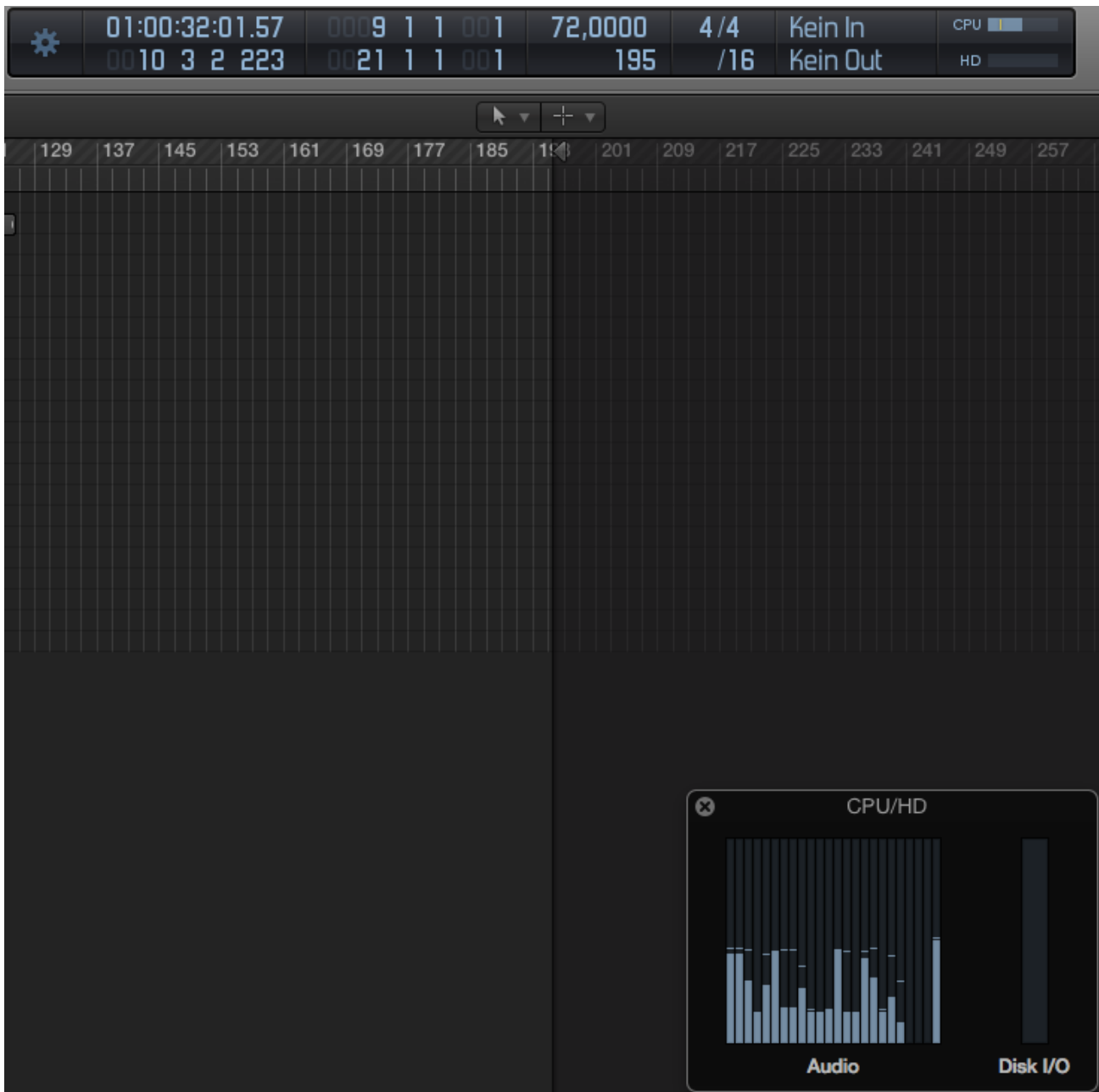
Habe auch dieses kleine Problem und es wirkt sich definitiv auf die Systemperformance aus.

Ich habe in Logic X ein Projekt geöffnet und auf die Auslastung geschaut einmal mit GenericUSBXHCI.kext und ohne.

Fürs erste habe ich die GenericUSBXHCI.kext entfernt um mit meinem System auch arbeiten zu können.

EDIT: FIX GEFUNDEN





Beitrag von „griven“ vom 24. Mai 2016, 23:29

Ich lese daraus, dass der GenericUSBXHCI.kext nicht gerade besonders sparsam mit den Ressourcen umgeht. Vielleicht ist hier eine Lösung über FakePCIID und FakePCIID_XHCIMux die bessere Wahl?

Beitrag von „apfelnico“ vom 25. Mai 2016, 01:12

In der Clover config.plist ein kext to patch:

Code

1. <dict>
2. <key>Comment</key>
3. <string>remove usb limit</string>
4. <key>Disabled</key>
5. <false/>
6. <key>Find</key>
7. <data>
8. g72M/v//EA==
9. </data>
10. <key>Name</key>
11. <string>AppleUSBXHCIPCI</string>
12. <key>Replace</key>
13. <data>
14. g72M/v//Fg==
15. </data>
16. </dict>

Alles anzeigen

und dann in den Clover kext-Ordner die Injector.kext rein (MacPro6.1 SMBIOS). Alle USB funktionieren damit bestens unter ElCapitan.

Edit: Die GenericUSBXHCI.kext natürlich raus

Beitrag von „DSM2“ vom 26. Mai 2016, 14:10

Oder einfach nur im Bios XHCI deaktivieren, EHCI aktivieren falls deaktiviert und GenericUSBXHCI.kext rein falls nicht mehr vorhanden.

DONE

Beitrag von „apfelnico“ vom 26. Mai 2016, 19:49

Vorteil Injector -> Es wird die Apple-eigene USB-Kext verwendet

Vorteil Clover-Patch -> Es wird beim Systemstart im RAM (die eigentliche Datei bleibt unberührt) die Apple-eigene USB-Kext derart verändert, dass das 15Port-Limit nicht mehr besteht und ALLE USB-Ports funktionieren.

Vorteil BIOS -> Es bleibt bei den Standardeinträgen, nicht unwichtig für weitere Betriebssysteme.

Beitrag von „DSM2“ vom 26. Mai 2016, 21:17

Ja, bei meiner Methode gibt es Schwierigkeiten wenn weitere Betriebssysteme verwendet werden,

da hast du natürlich recht aber für mich funktioniert es auch so, da meine DAW nur für OSX benutzt wird.

Habe deine Methode dennoch versucht aber endete leider dabei in ner Kernel Panic, was mir ehrlich gesagt die Stimmung verdorben hat auf Fehlersuche zu gehen.

Nachtrag:

```

panic(cpu 4 caller 0xfffff8000416d2f2): Kernel trap at 0xfffff800005954a, type 13=general protection, registers:
CODE SIGNING: cs_invalid_page(0x10b51f000): p=0[Header] Final status 0x200000, allowing (remove WRID) page
CR0: 0x0000000010003b, CR2: 0x0000000107ad000, CR3: 0x000000014be500, CR4: 0x00000000162000
RAX: 0xb0b78b40e2694055, RBX: 0xfffff8000b12f00, RCX: 0xfffff8000b14130, RDX: 0xfffff8000b0f000
RSP: 0xfffff8039773a60, RBP: 0xfffff8039773d70, RSI: 0xfffff8000b12f00, RDI: 0xfffff802ba7b22
R0: 0x000000000000000a, R9: 0x000000000000000a, R10: 0x0000000000000014, R11: 0xfffff804af4940
R12: 0xfffff802ba7b22, R13: 0xfffff80494c0300, R14: 0xfffff802ba7b22, R15: 0xfffff80407a0f00
RFL: 0x0000000000010206, RIP: 0xfffff800005954a, CS: 0x0000000000000000, SS: 0x0000000000000010
Fault CR2: 0x0000000107ad000, Error code: 0x0000000000000000, Fault CPU: 0x4

Debugger called: <panic>
Backtrace (CPU 4), Frame : Return Address
0xfffff8039773d610 : 0xfffff800032e021
0xfffff8039773d620 : 0xfffff8000416d2f
0xfffff8039773d630 : 0xfffff8000433ca3
0xfffff8039773d640 : 0xfffff800005954a
0xfffff8039773d650 : 0xfffff80192a614
hfs: mounted Studio on device disk1s3
0xfffff8039773d660 : 0xfffff800006522a
0xfffff8039773d670 : 0xfffff802ba7a641
0xfffff8039773d680 : 0xfffff8000063964
0xfffff8039773d690 : 0xfffff800006325f
0xfffff8039773d6a0 : 0xfffff80000640f3
0xfffff8039773d6b0 : 0xfffff80000640f3
0xfffff8039773d6c0 : 0xfffff8000416d2f

Kernel Extensions in backtrace:
com.apple.driver.X86PlatformPlugin(1.0)[12230E50-0C35-3365-8354-F696058F504100]0xfffff801929000->0xfffff801939fff
dependency: com.apple.iokit.IOUSBFamily(720.4.4)[4ED66649-76DE-3E91-9C38-322700047C7E100]0xfffff80c2000
dependency: com.apple.iokit.IOPCIFamily(1.1)[7E2095E-4918-3322-8188-75F063E8004C100]0xfffff801310000
dependency: com.apple.driver.IOPPlatformPluginFamily(5.7.1a6)[167BEDCC9-E9B0-3F32-82CE-04729702C0A100]0xfffff801930000
dependency: com.apple.driver.AppleSMB(3.1.9)[86F4D2C7-0200-3232-0C51-FFCE30291801100]0xfffff801930000
com.apple.driver.ApplePCI(1.7.3)[51E031C4-0376-3596-8C38-047180048F0E100]0xfffff802ba0000->0xfffff802ba00ff
dependency: com.apple.driver.IOPPlatformPluginFamily(5.7.1a6)[167BEDCC9-E9B0-3F32-82CE-04729702C0A100]0xfffff801930000
dependency: com.apple.iokit.IOPCIFamily(2.9)[20520CFE-FD28-3C37-9039-005201000025100]0xfffff800000000

BSD process name corresponding to current thread: kernel_task
Boot args: darwin=0 npci=0x2000 nvda_drv=1 kext-dev-mode=1 -f PCIrootUID=1 stld=0

Mac OS version:
14F27

Kernel version:
Darwin Kernel Version 14.0.0: Wed Jul 23 22:26:53 PDT 2015; root:xnu-2782.40.9~1/RELEASE_ARM64
Kernel UID: 0x0030545c7-3c7-8880-1738f01803c4
_NIB text base: 0xfffff8000100000
System model name: MacPro6,1 (Mac-0000000000000000)

System uptime in nanoseconds: 625208435
AppleIntelI1000(Info): e1000a NIC Link is Up 1000 Mbps Full Duplex, Flow Control: Rx/Tx

```

