

Erledigt

Anleitung - Intel Xeon E3 V2 oder V3 Prozessorgrafik aktivieren und verwenden

Beitrag von „griven“ vom 11. April 2016, 00:56

Wie Ihr vielleicht wisst verfügen einige Vertreter aus Intels E3 XEON Familie genau wie die "normalen" Core I Prozessoren über eine IGPU was für sich genommen schon mal praktisch ist, wenn man kein Grafikmonster braucht aber trotzdem nicht auf einen leistungsfähigen Prozessor verzichten möchte. Bisher war es jedoch immer eher schwierig die IGP´s der XEONs zur Zusammenarbeit mit OS-X oder aber besser gesagt OS-X zur Zusammenarbeit mit den in den XEON Prozessoren enthaltenen Grafikeinheiten zu bewegen. Hier nun eine kleine Anleitung wie Ihr die IGPU der E3-V2 oder E3-V3 Prozessoren unter OS-X trotzdem nutzen könnt.

1. Worin unterscheiden sich die Intel HD4000 und HD4600 von der P4000 oder P4600?

Im Grunde unterscheiden sie sich gar nicht denn technisch gesehen werfelt hier in beiden Chips der selbe Grafikcontroller nur eben unter unterschiedlichem Namen und mit einer anderen DeviceID letztlich sind die IGPU´s der XEON und Core I Prozessoren also identisch.

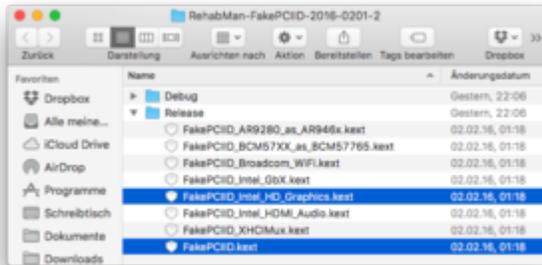
2. Warum funktionieren sie dann unter OS-X nicht?

Apple hat in keinem seiner Produkte jemals einen E3 Xeon verwendet sondern immer die größere E5 Linie die keine IGPU besitzt. Da die IGPU´s der E3 XEON´s eine andere DeviceID haben als die der Core I Prozessoren werden sie von OS-X schlicht ignoriert sprich die Grafiktreiber für die IntelHD Grafik werden nicht geladen da es für das OS so aussieht als sei keine unterstützte IGPU im System vorhanden. Schuld an diesem Verhalten ist die von den Core I Prozessoren abweichende DeviceID der IGPU denn diese ist schlicht nicht als Match für die Grafiktreiber hinterlegt und so kümmert sich OS-X eben nicht drum.

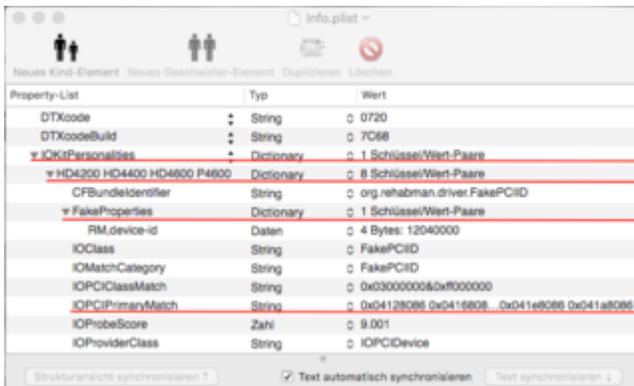
3. Und was muss ich nun tun damit es geht?

Im Grunde ist es ganz einfach denn es nicht wirklich mehr zu tun als OS-X für die Grafikkarte der XEON Prozessoren eine passende DeviceID vorzugaukeln und hier kommt einmal mehr die Arbeit von RehabMan ins Spiel denn dessen FakePCIID.kext kann man sich prima zu Nutze machen um OS-X eine passende Device ID zu servieren. Als Basis verwenden wir jeweils die letzte Version der FakePCIID.kext von RehabMan welche Ihr hier findet: [KLICK](#). Einmal heruntergeladen und entpackt findet Ihr in Eurem Download Ordner einen Ordner der die beiden Unterordner DEBUG und RELEASE enthält. Für unser weiteres Vorgehen verwenden wir

den Release Ordner und kopieren uns von dort 2 Dateien auf den Desktop:



Die FakePCIID.kext lassen wir in Ruhe aber bei der FakePCIID_Intel_HD_Graphics.kext machen wir einen Rechtsklick und wählen Paketinhalte zeigen und navigieren so dann in den Ordner Contents hier angekommen finden wir eine Info.plist vor, die wir nun mit einem plist Editor unserer Wahl öffnen (ich benutze plisteditpro es geht aber auch Apples XCODE oder ein Texteditor). Die für uns wichtigen Punkte zeigt das folgende Bild:

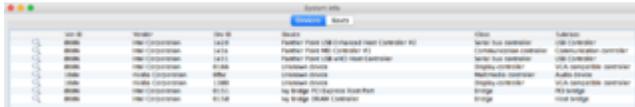


Nun geht es ans editieren der plist. Als erstes knöpfen wir uns den Punkt FakeProperties vor denn hier kommt der Wert rein, den wir OS-X unterschieben möchten. In meinem Fall war dies die DeviceID der HD4000. Der Wert liegt hier als 4Bytes Wert vor also muss man hier ein wenig um die Ecke denken aber erstmal muss man natürlich die richtige Device ID für die gewünschte Karte herausfinden aber glücklicherweise gibt es [hier](#) eine recht komplette Liste. In meinem Fall hat die IntelHD4000 also die Device ID 0x0166 und genau diese gilt es nun in die plist einzubauen. Auf der sicheren Seite sind die von uns die PlistEditPro verwenden denn hier ist es recht einfach den Wert zu ändern. PlistEditPro zeigt hier in der unbearbeiteten Fassung folgendes an:



Im Falle von PlistEditPro brauchen wir den Wert nur zu drehen sprich aus 0166 wird 6601 also

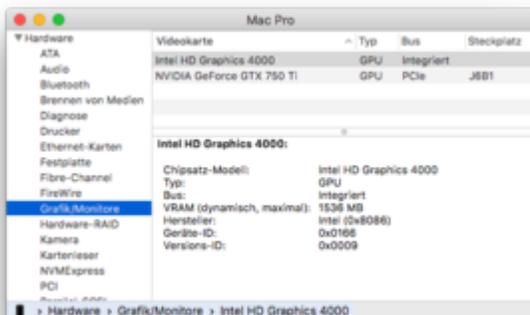
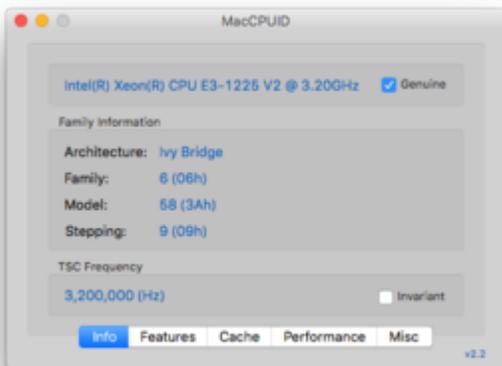
Bitweise verdreht demnach wird also aus 1204000 dann 6601000 und das tragen wir hier nun ein (Für die P4600 muss es 1604000 sein). Weiter geht es nun mit dem IOCPPrimaryMatch Property denn irgendwie müssen wir dem FakePCIID ja auch sagen worauf er anspringen soll. Dazu gilt es erstmal herauszufinden wie unsere DeviceID eigentlich lautet und hierzu eignet sich das Tool SystemInfo (gibt es im DL Bereich) prima. Einmal gestartet sieht der Output wie folgt aus:



Interessant an der Stelle ist die DeviceID des VGA Compatible Controller von Intel die wir uns dann auch merken (in meinem Fall war das die 016a). Um nun unsere DevcelID entsprechend eintragen zu können löschen wir zunächst alles, was unter IOCPPrimaryMatch steht sieht dann so aus:



Abschließend noch korrekt ausfüllen wobei hier gilt DEVICEIDVENDORID oder in meinem Fall 0x16a8066 also 0x16a für das Device und 8066 für den Vendor also Intel. Wenn alles erledigt ist speichern wir die bearbeitete plist ab und installieren dann beide Kexts mit dem Kextutility. Das Ergebnis nach einem Neustart sieht dann wie folgt aus:



Die XEON iGPU läuft und wird erkannt 😄

Beitrag von „Blade86“ vom 9. Mai 2016, 03:29

Hallo liebe Freunde!

Könntet Ihr mir vielleicht mit der Installation/Konfiguration einer Intel HD Graphics P4000 helfen?

Sie läuft unter 016a8086 also 016a device id und 8086 Geräte id.

Egal was ich mache ich bekomme nur 8MB als grafikspeicher und er lädt keine kext.

Hatte mal die Anleitung von griven probiert, leider erfolglos.

Hab versucht via clover inject fakeid und platform id was hinzubekommen, aber entweder ich hatte 8MB oder er blieb beim apple-logo mit progressbar stehen.

Ihr könnt alle infos von mir haben oder auch einen fernzugang bekommen.

Falls das einer von euch hinkriegen würde, würdet ihr mich ÜBERGLÜCKLICH machen 😊

Voller Hoffnung und Dankbarkeit

Blade

Beitrag von „al6042“ vom 9. Mai 2016, 07:13

Moin Blade86 und herzlich Willkommen im Forum... 😊

Ich habe deinen Post mal in den richtigen Thread verschoben, denn immerhin kämpfst du mit der IntelHD P4000 und nicht der IntelHD 4400... 😊

Beitrag von „Blade86“ vom 9. Mai 2016, 16:53

Herzlichen Dank [@al6042](#)!!

Ich muss wirklich sagen, dass dies eine außerordentlich freundliche Community ist. Die meisten Communities sind Neulingen gegenüber eher genervt eingestellt, ganz nach dem motto "muss der nap doch wissen". Das ist aber hier garnicht so!! Sorry wegen der Mühen des Verschiebens.

Liebe Grüße aus Köln 😊

Blade

[@griven](#) Hast du damals einen Monitor via DisplayPort angeschlossen?

Weil ich hab weder HDMI noch DVI, nur VGA und DP. Und da ich weder einen DP Monitor noch einen DP-Adapter hier habe, bin ich momentan via VGA dran. Könnte es daran liegen?

Hab den preallocated memory auf 64 MB runtergesetzt, jedoch immer noch keine Änderung.

EDIT: Einen Schritt weiter 😊

Beitrag von „griven“ vom 10. Mai 2016, 09:29

Sehr gut 😄

Wie bist Du nun vorgegangen?

Beitrag von „Blade86“ vom 11. Mai 2016, 19:35

Lieber griven 😊

Also man kann mit clover ja die DeviceID faken. Das ersetzt den kompletten Part deiner

Anleitung bezüglich "FakePCIID".

Also durch das setzen folgender Einstellungen in Clover Configurator:

ACPI:

- New Way -> Fix IntelGFX

- IntelGFX

Devices:

- FakeID-IntelGFX: 0x01668086

Graphics:

- Inject Intel

- ig-platfrom-id: 0x0166000*

Hier muss man mehrere ausprobieren. Meine results für das Sternchen*:

0 ja

1 ja

2 nein

3 ja

8 ja

9 ja

A nein

B nein

So danach wird die Karte wie in meinem Screenshot erkannt, ABER keine transparente dock menu bar und sehr langsame Animationen und rumgeflackere beim einloggen sind nach wie vor vorhanden.

Durch das Injecten hat mein Bootvorgang sehr viel länger gedauert.

hab dann den AppleIntelSNBGraphicsFB.kext gelöscht und den AppleIntelFramebufferCapri.kext modifiziert:

Hab herausgefunden dass man unbedingt Intel MEI braucht, was normalerweise auf 1E3A:8086 läuft, aber bei unserer Karte leider mit 1C3A:8086 angesprochen werden muss.

Deshalb habe ich die AppleIntelFramebufferCapri.kext so modifiziert, dass sie auf unsere ID anspringt, und BOOOM wieder einen schritt weiter.

Results:

-Bootvorgang wieder nur 5 Sekunden statt 5 Minuten

-1536MB grafik ram

-Intel MEI via AppleIntelFramebufferCapri.kext springt an.

Alles schön und gut, "ABER keine transparente dock menu bar und sehr langsame Animationen

und rumgeflackere beim einloggen sind nach wie vor vorhanden."

Sorry wegen der verspäteten Antwort, aber vor lauter kexts und clover settings und rebuilding und modden, wusste ich erstmal nicht welcher von meinen 10000Schritten ausschlaggebend für den bisherigen Erfolg war und musste erstmal alles sortieren, reproduzieren und Testen&verifizieren, bevor ich als total-noob irgendeinen Schwachsinn hier reinschreibe. (meistens geschehen die besten Fortschritte zu Uhrzeiten und in Zuständen, in denen ich besser keinen Text verfassen sollte)

Ich hoffe, dass ich mit diesen Infos dem ein oder anderen P4000 user helfen kann.

Euer Blade