

Erledigt

Wie wichtig ist die iGPU im Hacki und gibt es eine iGPU in einem echten Mac?

Beitrag von „redbelt“ vom 4. Dezember 2018, 15:00

Hallo zusammen,

ich würde mal gern grundlegend wissen wollen, ob die iGPU im Hack irgend ein Einfluss auf die Geschwindigkeit hat. Für den täglichen normalen Gebrauch. Laut Cinebench gibt es da keinen Unterschied. Werder GPU noch CPU.

Die iGPU wird wahrscheinlich nur dann wichtig wenn das Programm auch dementsprechend darauf zugreifen muss.

Da stellt sich mir gleich die Frage ob ein richtiger Mac eine iGPU (ist ja auch ein Intel CPU) verbaut hat oder ob der Mac nur auf die GPU der Grafikkarte zugreift?

Die GPU der Grafikkarten sollten eigentlich stärker sein als die iGPU des CPU's. Macht man zwischen den beiden ein Benchmark, ist die Grafikkarten GPU definitiv besser.

Wie verhält sich das aber mit einem echten Mac. Gibts da überhaupt eine iGPU? Ich kenne nur die eGPU.

Am liebsten wäre mir, die iGPU zu verbannen und nur hauptsächlich mit der Grafikkarte arbeiten. DRM oder Quicksync. Egal.

Unter Windoof wäre das ein Treiber Problem. 😊

Beitrag von „Si Vis Pacem“ vom 4. Dezember 2018, 15:30

Ähem, vielleicht täusche ich mich, warst du nicht in einigen diesbezüglichen Threads unterwegs?

Davon abgesehen ist es sehr schwierig Beiträge zum Thema Grafikkarte zu lesen ohne mit iGPU und HW-Beschleunigung konfrontiert zu werden.

TLDR:

Video wird idR mit iGPU schneller abgewickelt als ohne bzw. reduziert Systemlast.

Beitrag von „redbelt“ vom 4. Dezember 2018, 16:02

Ich befasse mich viel mit diesen speziellen Thema.

Hier und da habe ich schon was dazu geschrieben aber ehrlich gesagt, ich verliere gerade selber den Überblick.

Die Frage ist aber noch nicht von mir gekommen.

Mich interessiert es gerade bei echten mac, wie das dort gehandelt wird.

Beitrag von „CMMChris“ vom 4. Dezember 2018, 16:13

Was meinst du warum es je nach System so wichtig ist (gerade unter Mojave) die iGPU richtig eingebunden zu haben? Natürlich wird die iGPU bei einem echten Mac genutzt wenn sie vorhanden ist.

Was du offenbar nicht verstehst ist, dass die iGPU in Kombination mit einer dGPU nicht für 3D oder andere Grafikberechnungen genutzt wird. Genutzt wird sie fürs Medien [De- und Encoding](#) um die CPU zu entlasten. Seit Mojave nicht mehr nur für Videos sondern auch für Bilder (z.B. in der Vorschau).

Im Optimalfall nutzt das System sowohl die dGPU als auch die iGPU in Kombination zum [De- und Encoding](#)

. So ist es bei einem echten Mac und das bringt in der Theorie am meisten Leistung. Bei vielen Hackintosh Setups läuft es allerdings darauf hinaus, dass nur die IGPU genutzt wird, weil macOS DGPU En- und Decoding strikt nur auf in echten Macs genutzten GPUs zulässt. Das System geht dabei derart strikt vor, dass selbst eine RX580 nicht genutzt wird, bis sie entsprechend dem Equivalent im iMac deklariert wird (Radeon Pro 580 statt Radeon RX580).

Beitrag von „Si Vis Pacem“ vom 4. Dezember 2018, 16:59

[Zitat von CMMChris](#)

Bei vielen Hackintosh Setups läuft es allerdings darauf hinaus, dass nur die IGPU genutzt wird, weil macOS DGPU En- und Decoding strikt nur auf in echten Macs genutzten GPUs zulässt. Das System geht dabei derart strikt vor, dass selbst eine RX580 nicht genutzt wird, bis sie entsprechend dem Equivalent im iMac deklariert wird (Radeon Pro 580 statt Radeon RX580).

[CMMChris](#)

Ist diese Restriktion auch auf SMBios-Ebene? Bis jetzt war mein Eindruck, dass CPU -> SMBios das alleinige Kriterium war.

Ich bin noch immer am Suchen, wieso meine R9 390 von ca. 88 in Cinebench auf 73 runter ist. Glaube seit einer HS Beta.

Dies würde es dann erklären, da ich eine Zeitlang mit 5k late 2015 unterwegs war und jetzt mit 14,2

Beitrag von „Harper Lewis“ vom 4. Dezember 2018, 18:06

Wie äußert es sich denn genau, wenn die Eigenschaft *model* von *Radeon RX580* in *Radeon Pro 580* geändert wird? Macht sich das irgendwie im IORegistryExplorer bemerkbar?

Beitrag von „CMMChris“ vom 4. Dezember 2018, 19:08

[Harper Lewis](#) an dieser Stelle ist die liebe [macdesignerin](#) die richtige Ansprechpartnerin.

[Si Vis Pacem](#) Beim SMBIOS geht es hauptsächlich um die CPU und einige ACPI Gerätebezeichnungen. Auf GVA (Graphics Video Acceleration) hat das SMBIOS nur in wenigen Fällen Auswirkungen.

Beitrag von „redbelt“ vom 4. Dezember 2018, 19:18

Bringt mich irgendwie nicht weiter. Ja ja...verstehen tue ich das schon aber warum ist das so schwer alles miteinander zu koppeln?

Mehr als iGPU einschalten, kann ich doch nicht. Somit ist doch die iGPU und die dGPU vorhanden und sollte doch harmonisch miteinander arbeiten können.

Theorie und Praxis. Verdammt....

Beitrag von „ebs“ vom 4. Dezember 2018, 19:47

Z. B. mein MacBook Pro 2012 hat eine iGPU Intel HD Graphics 4000 und eine GeForce GT 650M mit automatischer Umschaltung zwischen den Grafikprozessoren. Es gibt auch ein nettes kleines Programm gfxCardStatus. Damit kann ich manuell festlegen welche Grafik vorzugsweise arbeiten soll, ist unterwegs für das sparen von Energie sehr gut wenn man sich auf die HD4000 festlegen kann. Auch zeigt das Programm an wenn automatisch umgeschaltet wird, z. B. wenn ich Photoshop nutze oder einen externen Bildschirm anschließe. Ohne das Programm entscheidet das System ganz alleine nach Leistungsanforderung welche Grafik-GPU aktiv wird.

Das ist bei allen Mac's mit iGPU und dezidiertem GPU so. Die neueren Mac's laufen zur Energieeinsparung viel länger mit iGPU, die aber auch leistungstärker ist als bei meinen alten Teil, und nur wenn es für Programme erforderlich ist wird umgeschaltet. Hier gibt es aber auch nicht mehr ein funktionierendes Programmchen, s. o., zum manuellen umschalten.

Das Problem das es nicht so auf dem Hackintosh läuft wird wohl mit der von Clover emulierten EFI zusammenhängen. Bei den richtigen Mac's ist das ja schon fest in der EFI verankert, hardwaremäßig. Genauso wie es einen kleinen Chip für DRM und Co gibt.

Beitrag von „CMMChris“ vom 4. Dezember 2018, 21:27

[ebs](#) Hier geht es nicht um Laptops wo die iGPU mit dem Monitor verbunden ist, sondern um Desktops, wo sie connectorless eingebunden ist.

Beitrag von „griven“ vom 4. Dezember 2018, 23:14

Nun sie werkeln ja auch harmonisch zusammen redbelt wenn eben die Voraussetzungen dafür geschaffen wurden. Es ist eben ein Trugschluss zu denken das SMBIOS hätte nur was mit der Optik zu tun bzw. war das ganz früher mal so aber in der Zwischenzeit hat sich einiges getan



Mal ein konkretes Beispiel:

Ausgehend von Deiner CPU würde man wohl als Modell am ehesten den iMac 18.3 wählen wollen denn das ist ein ziemlich perfektes Abbild von Deinem Setup. Die CPU ist identisch (i7-7700K) und auch die Grafik ist nahezu identisch RX580 vs. Radeon Pro 580 besser geht es SMBIOS technisch gar nicht. Gucken wir dem Bruder Deines Systems mal unter Die Haube bzw. hinter das Display. Wir haben also einen iMac mit der BoardID Mac-BE088AF8C5EB4FA2 das ist gut zu wissen denn damit bewaffnet können wir mal in der AppleGraphicsDevicePolicy gucken was der so können sollte. Der iMac 18.3 wird mit der config3 gelistet und die wiederum besagt

folgendes:

Code

```
1. <key>Config3</key>
2. <dict>
3. <key>GFX0</key>
4. <dict>
5. <key>EDID</key>
6. <dict>
7. <key>index</key>
8. <integer>0</integer>
9. </dict>
10. <key>FeatureControl</key>
11. <integer>12</integer>
12. <key>unload</key>
13. <false/>
14. </dict>
15. <key>IGPU</key>
16. <dict>
17. <key>unload</key>
18. <true/>
19. </dict>
20. </dict>
```

Alles anzeigen

Wir wissen nun das Gerät hat 2 Grafikeinheiten einmal die dGPU und die iGPU ferner wissen wir Apple erwartet das die iGPU im ACPI als solche zu finden ist und es erwartet wird das die dGPU als GFX0 im System vorhanden ist. Nun ist es bei nicht Apple Geräten oft so das die iGPU als GFX0 definiert ist im ACPI und die dGPU sich unter PEGP oder PEG verbirgt. Genau wissen wir das in Deinem Fall aber erst wenn wir einen Blick in die DSDT gewagt haben. Wichtig ist jedenfalls schon mal das wir diese, von Apple verlangte, Nomenklatur einhalten ansonsten führt das unweigerlich zu Chaos.

Was sagt uns unser Blick in den Kext noch nun er sagt uns das für die IGPU der Key `unload` auf `true` gesetzt wird und das meint nichts anderes als das der Framebuffer für die iGPU nicht geladen bzw. wieder entladen werden soll. So weit so gut denn den Framebuffer (Bildschirmtreiber für die Darstellung von 2D Inhalten auf Bildschirmen) brauchen wir ja auch gar nicht denn das übernimmt ja sowohl im iMac als auch in Deinem Fall die AMD Karte. Unser erstes Arbeitspaket lässt sich also hieraus ableiten:

1. In die DSDT gucken und herausfinden wie die Grafikkarten im ACPI angesprochen werden.
2. Dafür sorgen das dies macOS konform geschieht entweder über Clover oder direktem DSDT Patch.
- 2a. Wenn über Clover dann darauf achten das man als erstes GFX0 in IGPU umbenennt und erst dann zum Beispiel PEGP in GFX0.
3. Mittels Clover oder DSDT einen Connectorless PlattformID injecten (0x59120003).
4. Dafür Sorge tragen das es ein DEVICE IMEI gibt entweder durch umbenennen des vorhandenen MEI Devices oder durch einfügen eines Fake Devices.

Stellt sich jetzt natürlich die Frage warum der ganze Driss mit dem Apple konformen umbenennen im ACPI und die Antwort darauf ist relativ einfach! MacOS ist nicht Windows und hat für alles und jeden einen Treiber und es ist auch nicht Linux das sich für jede Vendor und DeviceID schon was passendes aussucht sondern macOS erwartet einen Mac und geht daher davon aus die nötigen Informationen zum einen aus dem SMBIOS zu ziehen zum anderen aber auch aus dem ACPI und eben so sicher wie es davon ausgeht dort die entsprechenden Informationen zu finden macht es davon auch abhängig welche Funktionen auf der erkannten Hardware bereit gestellt werden (AirPlay, H264 en- und decoding, HVEC en- und decoding usw.)

Nachdem wir das erledigt haben sind wir dem Ziel schon ein gutes Stück näher gekommen denn unser nicht ganz echter iMac kennt nur bestenfalls seine iGPU und auch eine (bewusst nicht seine) dGPU und kann zumindest schon mal mit den Informationen über die iGPU arbeiten. Warum schreibe ich eine und nicht seine dGPU auch das ist wieder relativ leicht zu erklären denn der Apple verwendet ja nichts von der Stange sondern die Polaris Karten in den Mac's sind nicht irgendwelche Polaris Karten (haha) sondern es sind Radeon Pro Karten. Der Fachmann staunt und der Laie wundert sich was macht nun eine Brot und Butter Polaris Karte zu einer Radeon Pro und was zum Teufel ist Radeon Pro überhaupt?

Hier kommt wieder der von [CMMChris](#) bereits erwähnte Fakt ins Spiel. Apple ist auf die glorreiche Idee gekommen so genannte "egpu developer kits" zu verkaufen in denen eine AMD RX verbaut war und die es Entwicklern ermöglichen sollte schon mal an der Leistungsfähigkeit der Metal API sowie der kommenden dGPU Generation zu naschen ohne dafür konkret einen Mac in Petto zu haben der das bereits verbaut hatte. Jetzt hat man also eine ganze Horde von

Entwicklern die sich die Grafikkarte gekauft haben und wie überzeugt man die nun davon das Dingen wieder in die Ecke zu stellen und einen hochpreisigen Mac zu kaufen ach eigentlich ganz einfach man sparrt sich ein paar wenige, aber essentielle, Funktionen auf die sich nicht über Thunderbolt tunneln lassen, lötet die gleiche GPU auf das LogicBoard verpasste dem Dingen aber vorsichtshalber vorher eine andere DeviceID und zack hat man eine Radeon Pro ersonnen. Der Fachmann schüttelt den Kopf und kauft trotzdem denn gibt ja keine Alternative und der Laie wundert sich schon wieder und huldigt Apple oder so ähnlich...

Was sagt uns das nun? Die RX580 ist genau so Pro wie die Radeon Pro 580 man muss macOS das nur verklickern und das geht indem man der die RX entsprechend verkleidet zum Beispiel hiermit: [AMDRadeonPro.kext.zip](#) wobei es sich hierbei um einen alten Bekannten handelt nämlich um den PropertyInjector.kext von [Brumbaer](#). Natürlich kannst Du das Dingen jetzt nicht einfach reinwerfen und hoffen das es geht vielmehr musst Du in der info.plist des Kexts die Device und VendorID unter PCIPrimaryMatch noch an Deine Gegebenheiten anpassen aber ist das einmal getan steht dem fast echten iMac 18.3 nichts mehr im Wege.

Puh nun ist es mehr Text geworden als ich eigentlich schreiben wollte aber vielleicht hilft das ein wenig Licht ins Dunkle zu bringen...

Beitrag von „CMMChris“ vom 5. Dezember 2018, 01:19

Besser kann man es nicht erklären. Der Text sollte an passender Stelle in die Wiki.

Beitrag von „redbelt“ vom 5. Dezember 2018, 14:27

[griven](#)

Phuuuu...Respekt! Also erst mal Vielen Dank für den kurzen - langen Überblick, was man tun muss um einen "echten" Mac zu bekommen.

Das umsetzen wird für mich die höchste Hürde sein.

Das geht schon los, in welcher DSDT nach gucken muss und was ich suchen muss.

Du schriebst: entweder über Clover oder direktem DSDT Patch. Ist das egal oder kann man in einem Patch mehr machen als in Clover selber?

Wenn ich die GFX0 in IGPU umbenenne und die PEGP in GFX0, wie kriege ich die richtigen FIND und REPLACE Daten raus?

Bei DEVICE IMEI ist eingetragen in der DSDT in Clover. Mittels Auswahlliste. Bei der Fake Device muss ich auch erstmal wissen, was da rein soll.

Weiterhin schriebs du: Natürlich kannst Du das Ding jetzt nicht einfach reinwerfen und hoffen das es geht vielmehr musst Du in der info.plist des Kexts die Device und VendorID unter PCIPrimaryMatch noch an Deine Gegebenheiten anpassen aber ist das einmal getan steht dem fast echten iMac 18.3 nichts mehr im Wege.


Die ist schon für die 580 umgeschrieben, dank [macdesignerin](#) oder muss ich da noch mehr ändern?

Das ganze bring mich jetzt dahin, mit viel tiefer mit der Materie zu beschäftigen. Je mehr ich weiss um so mehr lerne ich und werde besser.

Dank dir [griven](#) und dem gesamten Forum.

Ich hoffe, dass ich jetzt weiter komme.

Beitrag von „coopter“ vom 5. Dezember 2018, 16:13

Die Karte (Sapphire RX 580 Nitro+).. meines Sohnes, zeigt ihm unter Sierra an, das es eine Radeon RX 580 ist ! Sollte da eigentlich Radeon Pro 580 stehen ? 

Beitrag von „redbelt“ vom 5. Dezember 2018, 16:19

Ja. Mit dieser kext.

Beitrag von „coopter“ vom 5. Dezember 2018, 16:28

@[redbelt](#)

Danke werde es Ihm sagen und testen 😊

Beitrag von „lucifer“ vom 5. Dezember 2018, 17:04

Also bei mir funktioniert der Kext nicht. Die RX580 wird ebenso als Rx580 erkannt, kein Hardwarerendering.

Seltsamerweise wird aber auch WEG nicht geladen, egal wo der liegt, ob in der EFI oder in L/E, S/L/E hab ich nicht probiert.

10.14	02.08.2018, 14:11	--	Ordner
Other	Heute, 16:29	--	Ordner
AMDRadeonPro.kext	25.07.2018, 20:28	49 KB	Kernel-...iterung
FakeSMC.kext	17.11.2018, 00:03	125 KB	Kernel-...iterung
Lilu.kext	30.10.2018, 12:01	148 KB	Kernel-...iterung
NoVPAJpeg.kext	22.11.2018, 22:20	31 KB	Kernel-...iterung
WhateverGreen.kext	30.10.2018, 11:08	425 KB	Kernel-...iterung
WhateverGreen.kext.dSYM	30.10.2018, 11:08	488 KB	Paket
XHCI-300-series-injector.kext	08.10.2018, 14:48	1 KB	Kernel-...iterung



WhateverGreen:

Version:	1.2.4
Zuletzt geändert:	30.10.18, 11:08
Paket-ID:	as.vit9696.WhateverGreen
Geladen:	Nein
Erhalten von:	Nicht signiert
Art:	Intel
Architekturen:	x86_64
64-Bit (Intel):	Ja
Ort:	/Library/Extensions/WhateverGreen.kext
Kext-Version:	1.2.4
Ladbar:	Nein
Fehler bei der Unterschriftenvalidierung:	Nicht signiert
Abhängigkeiten:	Erfüllt
Signiert von:	Nicht signiert

Beitrag von „CMMChris“ vom 5. Dezember 2018, 17:22

Mit der Kext hatten einige Nutzer Probleme. Das ganze funktioniert wohl nicht zuverlässig. Ich habe auch schon versucht meine RX Vega 64 mit der Methode zur Radeon Pro Vega 64 zu machen - bisher ohne Erfolg.

Beitrag von „Harper Lewis“ vom 5. Dezember 2018, 17:34

Ich habe das Gefühl, dass VideoProc nicht wirklich zuverlässig bzw. aussagekräftig ist. Für meine R9 280 wird mir angezeigt, dass Hardwarebeschleunigung komplett aktiv wäre, auch HEVC-De und -Encoding.

Nachtrag: Im Kext werden u.a. die properties name und model gesetzt, beide auf AMD Radeon Pro 580. Auf einem iMac18,3 mit AMD Radeon Pro 570 sieht das hingegen so aus:

model: <"Radeon Pro 570">

name: <"ATY,FlorinParent">

Beitrag von „CMMChris“ vom 5. Dezember 2018, 17:38

Dann prüfe es doch nach. Musst ja nur was konvertieren und die GPU Auslastung beobachten.

Beitrag von „coopter“ vom 5. Dezember 2018, 17:42

Bleibt bei dem Radeon RX 580 Eintrag in der SysInfo.

Beitrag von „Harper Lewis“ vom 5. Dezember 2018, 17:47

[CMMChris](#) : Habe ich bereits. iStat Menues zeit beim [Encoding und Decoding](#) eines HEVC-Videos zwar an, dass der GPU-Prozessor ausgelastet ist, die CPU-Auslastung ist aber ebenfalls hoch. Meines Wissens ist das [De- und Encoding](#) von HEVC auch erst ab Polaris möglich.

Beitrag von „redbelt“ vom 5. Dezember 2018, 17:49

Bei mir sieht es so aus.

Beitrag von „CMMChris“ vom 5. Dezember 2018, 18:02

[Harper Lewis](#) Bei mir springt bei HEVC mit der Vega auch die CPU hoch. Arbeitest du mit IGPU?

Beitrag von „Harper Lewis“ vom 5. Dezember 2018, 18:03

Welcher Framebuffer wird denn geladen? `ioreg -p IODeviceTree -n GFX0 | grep Framebuffer`

Nachtrag: Beim Encodieren eines HEVC-Videos sind die CPU und GPU bei mir auf Anschlag, beim Encoding nur die GPU und die CPU taktet auch herunter. Die R9 280 scheint also HEVC dekodieren zu können, wobei in VideoProc auch das Encoding als aktiv angezeigt wird.

Beitrag von „Harper Lewis“ vom 5. Dezember 2018, 18:05

Ja, die IGPU ist connectorless eingebunden. Beim Abspielen eines HEVC-Videos ist die IGPU laut Intel Power Menu und iStat Menus nicht aktiv. Ich nutze IINA zum Abspielen.

Beitrag von „Si Vis Pacem“ vom 5. Dezember 2018, 18:12

Bezüglich der Kext:

Unter IOKitPersonalities > AMD Radeon Pro 580 > Properties gibt es einen Eintrag "typ". Ist der wirklich richtig? Frage, da der Rest ja auch Englisch ist und hätte da ein "type" erwartet.

Edit:

Wird der Eintrag Name nicht für den Framebuffer verwendet? Hm.

Beitrag von „elmacci“ vom 5. Dezember 2018, 18:16

[Zitat von CMMChris](#)

Mit der Kext hatten einige Nutzer Probleme. Das ganze funktioniert wohl nicht zuverlässig. Ich habe auch schon versucht meine RX Vega 64 mit der Methode zur Radeon Pro Vega 64 zu machen - bisher ohne Erfolg.

Dieser Kext von macdesignerin aus diesem Thread hat bei Dir nicht funktioniert?

Damit konnte ich meine Vegas erfolgreich als Radeon Pro 64 definieren. Nur gebracht hat es leider nichts. [#42](#)

Beitrag von „CMMChris“ vom 5. Dezember 2018, 18:43

Nö da tut sich bei mir gar nix [elmacci](#)

Der Kext an sich läuft aber, habe mal testhalber die PowerPlayTable als Property mitgegeben --> läuft. Nur der Name will nicht.

Achja, was meinst du mit "bringt nichts"? Das Vega Encoding geht doch sowieso mit allen Vegas, da ist der Name egal. Das Ding "Radeon Pro Vega 64" zu nennen ist nur Kosmetik.

Trotzdem würde mich mal interessieren warum sich bei mir der Name nicht ändern lässt...

Beitrag von „elmacci“ vom 5. Dezember 2018, 18:50

"Bringt nichts" im Sinne von h264 Encoding über die Vega wenn man 2 Vegas hat (und die Rahmenbedingungen stimmen, sprich 10.14.0 oder 10.14.2, IGPU aus etc.).

Die Idee von macdesignerin war ja, dass es einfach nur - analog der Radeon Pro 580 - nur am Namen liegt dass das Encoding nicht klappt.

Beitrag von „CMMChris“ vom 5. Dezember 2018, 18:52

Ah, ne denke nicht dass es damit zusammenhängt. Wahrscheinlich unterstützt der Treiber einfach keine zwei Vegas und das ist der Grund?! Wie gesagt, das Decoding läuft ja einfach so bei einer Vega und ich bezweifle, dass der Name vom iMac Pro da an der Dual Vega Situation was ändert. Im iMac Pro steckt ja auch nur eine Vega.

Beitrag von „elmacci“ vom 5. Dezember 2018, 19:04

Ja, Dual Vega ist halt einfach "Special" 😊 Wobei mich ja schon eine Tatsache irritiert, bei:

- deaktivierter IGPU
- macOS 10.14.0
- 2 x Vega (1x GFX0 und 1x GFX1)

funzt DRM nicht.

Wenn ich aber die eine Vega GFX0 benenne über eine SSDT und die andere statt GFX1 --> GFX2, dann habe ich auf einmal funktionierendes DRM. HEVC ist grün, aber h264 halt leider immer noch nicht.

Allein die Bezeichnung hat hier also schon eine Veränderung im Verhalten bewirkt...

Na ja, ich bin mal gespannt ob die (Dual)Vega-Experten DSM2 oder [apfelnico](#) hier weiterkommen.

Beitrag von „Karl_der_Kaefer“ vom 5. Dezember 2018, 19:23

Thema ist Erl

Beitrag von „DSM2“ vom 5. Dezember 2018, 19:34

[Zitat von CMMChris](#)

Wahrscheinlich unterstützt der Treiber einfach keine zwei Vegas und das ist der Grund?!

Nicht ganz. Am iMac Pro geht es mit zwei Vegas sehr wohl, dabei ist die zweite GPU aber nicht über eine PCIe Anbindung im System hinterlegt sondern per Thunderbolt als eGPU. Ein Riegel ist aber seitens Apple drin... Aktuell weiß ich nur leider nicht wie ich das ganze umgehen soll.

Bezüglich renamings ist das alles Quatsch wie du schon sagtest...

MacOS ist es völlig egal wie die Karte heißt, selbst wenn sie Fritz heißt.

Beitrag von „kaneske“ vom 5. Dezember 2018, 19:40

[Zitat von elmacci](#)

Wenn ich aber die eine Vega GFX0 benenne über eine SSDT und die andere statt GFX1
--> GFX2

Wie realisierst du das genau? Bin da kein SSDT Experte...

Beitrag von „CMMChris“ vom 5. Dezember 2018, 19:45

Zitat

Nicht ganz. Am iMac Pro geht es mit zwei Vegas sehr wohl, dabei ist die zweite GPU
aber nicht über eine PCIe Anbindung im System hinterlegt sondern per Thunderbolt als
eGPU.

Ist mir schon klar, ich bezog mich allerdings auf den iMac Pro Auslieferungszustand, an
welchem sich die Rename Methode orientiert (Name).

Beitrag von „elmacci“ vom 5. Dezember 2018, 20:47

[Zitat von kaneske](#)

Wie realisierst du das genau? Bin da kein SSDT Experte...

Über eine SSDT - hatte mir mal vor einiger Zeit auf Grundlage von apfelnicos SSDT für seine
Vegas eine eigene gebastelt.

Hier mal angefügt.

Ob die Devicepfade bei Dir funktionieren müsstest Du mal mit IOReg prüfen, ansonsten anpassen.

Die SSDT in ACPI/patched legen und Whatevergreen (falls genutzt) sowie ggf. die interne iGPU deaktivieren.

Beitrag von „CMMChris“ vom 5. Dezember 2018, 20:53

LOL! Nach dem 10.14.2 Update geht der Rename der Karte nun. Sachen gibt's...

Beitrag von „elmacci“ vom 5. Dezember 2018, 20:56

Haha, geil - immer wieder was neues 😊

Beitrag von „CMMChris“ vom 5. Dezember 2018, 21:19

Kommando zurück: 1 Reboot --> Rename läuft wieder nicht.

Edit: VideoProc zeigt nach dem Reboot nun wieder die iGPU statt die dGPU an. Direkt nach dem Update ohne Reboot hat es einmal die Vega angezeigt. Ich verstehe das nicht... Wenn ich die iGPU abschalte steht in VideoProc "N/A". GVA läuft aber trotzdem, sowohl HEVC als auch H.264.

Beitrag von „Si Vis Pacem“ vom 6. Dezember 2018, 00:12

Wer Probleme mit dem Rename bzw. FakeID in der SSDT hat:

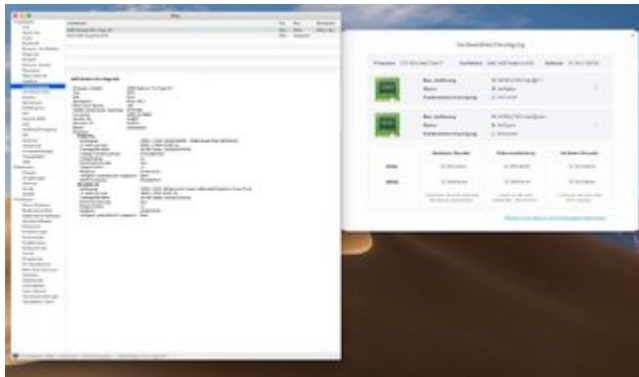
Mal mit "@0,model" probieren. Meine R9 390 hat auch immer rumgezickt bis ich den ersten Connector als Naming Convention benutzt habe.

Ditto für "device-id" und "vendor-id".

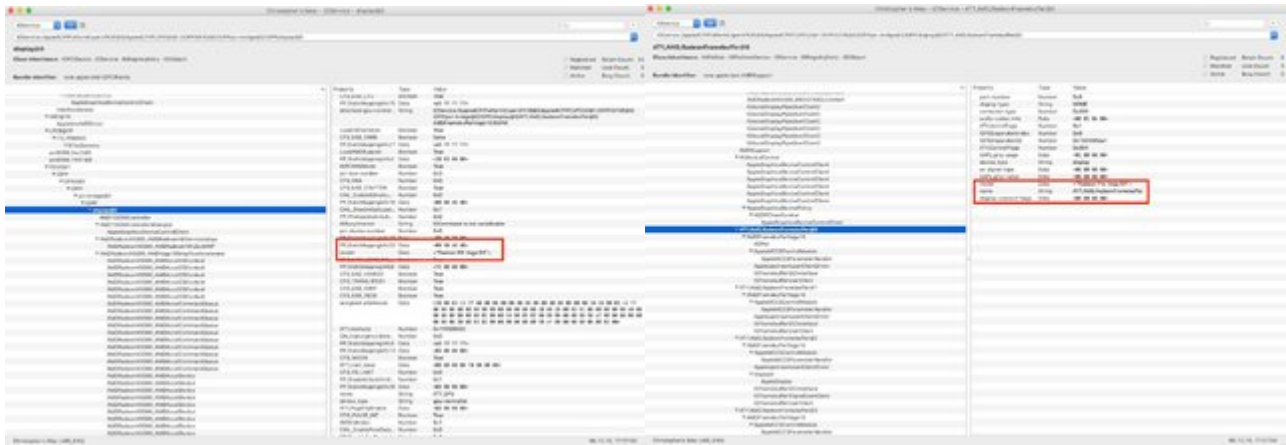
Sachen gibt's, die gibt's gar nicht 😊

Beitrag von „CMMChris“ vom 6. Dezember 2018, 16:47

Ja, @0,name tut es tatsächlich.



Ändert aber leider nichts an der Anzeige in VideoProc. Da hakt es also noch irgendwo anders. DSM2 hast du eine Idee wie es sein kann, dass nach dem Update alles korrekt angezeigt wurde und dann nach dem Neustart nicht mehr?



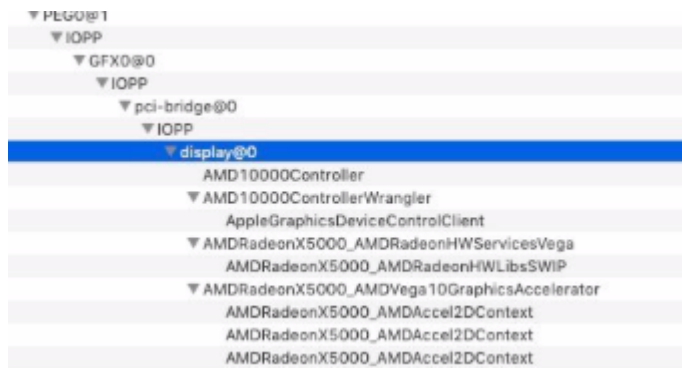
Der Rename greift nur auf den Framebuffer Einträgen und nicht auf dem Eintrag der Grafikkarte (display@0). Auch mit dem "delay" Parameter ist da nix zu machen.

Edit 2: Wenn ich Whatevergreen reinnehme greift der Rename auch auf display@0 bzw. dann GFX0 (WEG benennt das um). Leider keine Auswirkungen auf die Anzeige in Video Proc. Aber immerhin lässt sich so mit shikigva=1 die Nutzung der Vega forcieren. Anschließend läuft DRM, Vorschau geht auch noch und man hat bessere H.264 Performance.

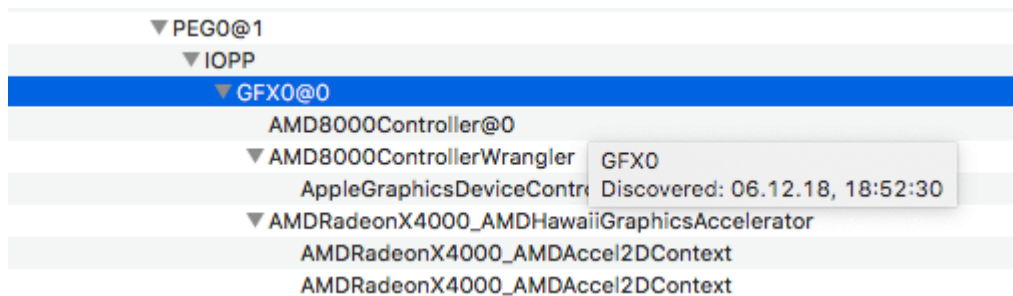
Beitrag von „Si Vis Pacem“ vom 6. Dezember 2018, 19:02

[CMMChris](#)

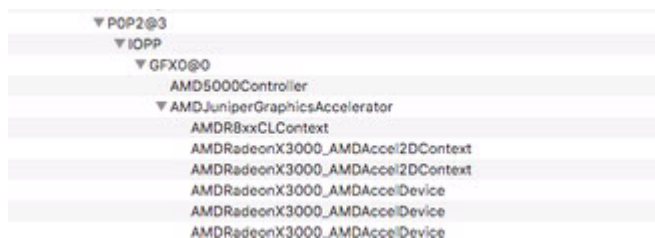
Habe - noch - keine Vega, aber soll das so aussehen?



Häcki



iMac



Du hast zweimal IOPP und dann display als device. Zumindest bei unserem 2010er iMac ist auch nur einmal IOPP.

Oder hat dies mit TB3 zu tun?

Edit:

Gerade aus einem anderen Grund von VirtualSMC zurück zu FakeSMC und nun zeigt mir iStat auch wieder die HD4000 an. Vielleicht mal zu FakeSMC zurück und vergleichen, da es doch Unterschiede zwischen den Beiden gibt.

Beitrag von „kaneske“ vom 6. Dezember 2018, 21:54

[Zitat von elmacci](#)

Ob die Devicepfade bei Dir funktionieren müsstest Du mal mit IOReg prüfen, ansonsten anpassen.

Das peile ich noch nicht ganz im MaciASL:

sind meine Device Pfade anders? Wenn ja was genau ist anders? Kann mir einer das erklären? Ist es bei mir der Pfad IOPP anstatt PEGP? Oder wie? Hä? Verstehe die Syntax da noch nicht.

Verstehe ich das richtig, dass ich den Scope auf den Pfad der Device triggere, also in der DSL Datei hier dann auf _SB.PCI0.PEG0 danach dann auf PEGP und EPG0 dann auf die Adresse _ADR und die Device GFX0???

Danke für Hilfe dabei.

Beitrag von „CMMChris“ vom 6. Dezember 2018, 22:26

[Si Vis Pacem](#) Ich arbeite an der Vega nicht mit einer SSDT sondern mit Clover oder dem PropertyInjector Kext. Bzgl. dem ACPI Pfad brauchst du dir keinen Kopf zu machen. Auf die Funktion hat das keine Auswirkung. Wichtig ist, dass PEGP zu GFX0 umbenannt ist. Wenn der

Pfad dem vom iMac gleichen soll musst du Whatevergreen reinnehmen. Wie man den Pfad per DSDT oder SSDT anpasst kann ich nicht sagen, kenne mich mit dem Kram nicht aus und mogele mich bisher erfolgreich drum herum. 😄
Edit: Elmacci hat's ja unter mir erklärt.

Beitrag von „elmacci“ vom 6. Dezember 2018, 22:26

Es kann sein dass Deine Device-Pfade anders sind, muss aber nicht.

Als Beispiel:

Wenn ich ohne Whatevergreen boote dann sehen die Pfade meiner beiden Vegas in IOREG folgendermaßen aus:

Vega 1: PEG0/PEGP/pci-bridge/display0

Vega 2: PEG1/PEGP/pci-bridge/display0

Jetzt nehmen wir die SSDT mal her:

Über den "Scope" teile ich dem Rechner erst einmal mit, in welchem bereits existierenden Bereich der Code arbeiten soll.

Das ist in meinem Fall PEG0 bzw. PEG1 und PEGP, deshalb findet sich auch in der SSDT jeweils unter "Scope":

Scope (_SB.PCI0.PEG0)

gefolgt von

Scope (PEGP)

In diesem Scope wird jetzt für den nächsten Unterpunkt (pci-bridge) ein neues Device "injiziert" - in meinem Fall EGP0 und EGP1 je VEGA.

Die kannst Du auch GPU1 und GPU2 nennen oder wie auch immer Du lustig bist 😊

Und schließlich wird für den letzten Unterpunkt statt "display0" GFX0 bzw. GFX1 injiziert.

Es kann sein dass Du in Deinem System unter PCI0 nicht 4 (PEG0, PEGP, pci-bridge, display) Punkte hast sondern bspw. nur 3.

In dem Fall musst Du die SSDT entsprechend anpassen, also bspw. einen Scope oder ein Device löschen.

Ich bin jetzt auch nicht der "SSDT-Crack" und hoffe ich habe es einigermaßen korrekt und verständlich wiedergeben können 😊

In den einzelnen GFX-Devices werden dann die Eigenschaften beschrieben, wie bspw. Name der Karte, Art des Framebuffers etc.

Ich nutze hier den Framebuffer "Iriri" da damit die Ports an meiner Grafikkarte auch mit allen Monitoren funktionieren.

Alternativ gibt es noch "Kamarang". Falls also ein oder mehrere Ports mit Iriri bei Dir nicht gehen dann mal mit Kamarang probieren.

Gibt auch noch "Japura", das ist aber der Framebuffer für die Vega im iMac Pro und macht meines Wissens nach keinen Sinn.

Beitrag von „kaneske“ vom 6. Dezember 2018, 22:47

Vielen Dank, das werde ich mal testen und berichten.

Beitrag von „DSM2“ vom 7. Dezember 2018, 12:10

[CMMChris](#) : Sorry jetzt erst die Zeit zu gefunden hier zu antworten. Ich habe mir das in der Hinsicht noch nicht angeschaut, bei renamings arbeite ich aber auch anders und zwar per Kext. Sobald man die devices hinterlegt (IOPropertyMatch) und zusätzlich den pcidebug hinterlegt

von der GPU, ändert sich da nichts aber der muss auch korrekt sein sonst bringt das nichts ...

Beitrag von „CMMChris“ vom 7. Dezember 2018, 14:14

Wie genau änderst du den Namen per Kext? Ich habe den Property Injector genutzt und es auch mit Clover versucht. Der Rename greift nicht. Erst wenn ich WEG dazu nehme wird der Name geändert. Aber wie schon erläutert: Die iMac Pro Bezeichnung für die Vega ändert nichts daran, dass VideoProc die falsche GPU anzeigt bzw. bei deaktivierter GPU "N/A" anzeigt. Es muss also einen anderen Grund gehabt haben, dass direkt nach dem 10.14.2 bis zum ersten Reboot die korrekte GPU in VideoProc angezeigt wurde.

Und falls die Frage aufkommt: Nein, die Anzeige ist nicht kosmetisch. Im VTDecoderXPCService werden die AMD Videobeschleuniger nicht aufgeführt, solange die GPU nicht korrekt in VideoProc erscheint. Ich bin mir ziemlich sicher, dass das auch der Grund dafür ist, dass wir hier anhaltende Probleme mit DRM bzw. Netflix in Safari haben. Es wäre für die gesamte Hackintosh Community von Vorteil, wenn man hier den Grund für das falsche Verhalten herausfinden könnte.

Beitrag von „DSM2“ vom 7. Dezember 2018, 16:40

Methoden gibt es mehrere aber mit propertyinjector Kext geht das wunderbar.

IOPropertyMatch und zusätzlich die pcidebug deiner Vega rein und dann sollte wenn die pcidebug korrekt ist, auch deine Karte mit dem von dir gewünschten Namen erscheinen ohne WEG.

[CMMChris](#)

Beitrag von „CMMChris“ vom 7. Dezember 2018, 17:00

Uhm, na genau davon rede ich doch oben. Funktioniert solange nicht bis WEG drin ist. Nur wenn ich wie oben von einem Nutzer beschrieben per "@0,model" den Namen setze, funktioniert es ohne WEG. In dem Fall ist aber der Rename an der falschen Stelle, d.h. im Framebuffer statt im GFX0 / display@0 Eintrag der Karte. Das führt dann zwar zur Anzeige als Radeon Pro Vega 64 in "über diesen Mac" und im System Profiler, überschreibt aber nicht den Kartennamen den sich diverse Apps ziehen. In iStat Menus wird dann z.B. nach wie vor "Radeon RX Vega 64" angezeigt. Der Property Injector Rename auf "name" und "model" für GFX0 / display@0 greift erst dann wenn WEG drin ist. Selbiges gilt für den Inject per Clover auf den PCI Pfad.

Das ändern des Namen der Karte zu "Radeon Pro Vega 64" hat aber auf die Funktion keinerlei Auswirkungen und ist rein kosmetisch. Deshalb ist mir das mit dem nicht funktionieren Rename erstmal total egal. Viel eher interessiert mich das in Beitrag #49 Beschriebene.

Beitrag von „DSM2“ vom 7. Dezember 2018, 17:36

Bist du dir ganz sicher das den "richtigen" debug wert hast?

Beitrag von „CMMChris“ vom 7. Dezember 2018, 17:37

Ja, sonst würde ja das Injecten nicht funktionieren 😊

Beitrag von „DSM2“ vom 7. Dezember 2018, 17:41

Für einen einfachen inject würde die IOPropertyMatch ausreichen ohne korrekten pcidebug Wert ... würde aber genauso springen wie aktuell 😊

Beitrag von „CMMChris“ vom 7. Dezember 2018, 17:56

Springen? Wo springt was? Verstehe nicht was du damit sagen willst. Wie gesagt, PCIDebug Wert stimmt und ich Injecte auf die PCI ID. Ich kann auch ohne Kext mit Clover auf den exakten PCI Pfad injecten. Die Werte werden ja auch alle angenommen (PowerPlayTable, DisableFanControll, Framebuffer Injects), nur "name" schluckt er erst dann wenn WEG drin ist.

Beitrag von „DSM2“ vom 7. Dezember 2018, 18:33

Mit springen meine ich wenn er vorher bei dir den Namen genommen hat und dann doch wieder den injected Namen nicht lädt.

Strange, würde es jetzt testen aber mein X299 ist grad in Einzelteile verstreut...

Beitrag von „CMMChris“ vom 7. Dezember 2018, 18:55

Sorry aber wo habe ich geschrieben dass er den Namen mal laden würde und mal nicht? Das ganze ist nur ein einziges Mal geschehen: Nach dem Update auf 10.14.2. Nach einem Reboot war es dann wieder weg. Zeitgleich wurde auch in VideoProc die Vega angezeigt. Auch das war nach dem ersten Reboot dann wieder weg.

Am funktionierenden Rename lag es aber nicht! Der Rename zu Radeon Pro Vega 64 hat keine Auswirkung darauf, ob die Vega in Video Proc auftaucht oder nicht. Letzteres ist auch das, was für mich wichtig ist, da es darüber entscheidet ob DRM ohne Umwege funktioniert und die Vega bei H.264 Encoding ordentlich ausgelastet wird. Leider habe ich bisher keinerlei Anhaltspunkt der mich zur Wurzel des Problems führen würde.

Nochmal zurück zum Rename: Ich Wiederhole nochmal die Fakten, vielleicht wird es dann klarer:

Inject via CLOver auf den PCI Pfad (Device Properties)

Ohne WEG

- name = Radeon Pro Vega 64 -> keine Auswirkung
- model = Radeon Pro Vega 64 -> keine Auswirkung
- @0,model = Radeon pro Vega 64 -> Radeon Pro Vega 64 im System Profiler, nicht jedoch in Apps. Model Inject befindet sich im Framebuffer Eintrag. In GFX0 bzw. display@0 steht nach wie vor "Radeon RX Vega 64".
- Inject von PowerPlayParametern etc. funktioniert problemlos.

Mit WEG

- name = Radeon Pro Vega 64 -> Radeon Pro Vega 64 im System Profiler und in Apps.
- model = Radeon Pro Vega 64 -> Radeon Pro Vega 64 im System Profiler und in Apps.

Inject per Property Injector mit korrekter PCIID und PCIDebug

Ohne WEG

- name = Radeon Pro Vega 64 -> keine Auswirkung
- model = Radeon Pro Vega 64 -> keine Auswirkung
- @0,model = Radeon pro Vega 64 -> Radeon Pro Vega 64 im System Profiler, nicht jedoch in Apps. Model Inject befindet sich im Framebuffer Eintrag. In GFX0 bzw. display@0 steht nach wie vor "Radeon RX Vega 64".
- Inject von PowerPlayParametern etc. funktioniert problemlos.

Mit WEG

- name = Radeon Pro Vega 64 -> Radeon Pro Vega 64 im System Profiler und in Apps.
- model = Radeon Pro Vega 64 -> Radeon Pro Vega 64 im System Profiler und in Apps.

Inject per Property Injector mit korrekter PCIID und PCIDebug + Delay (getestet mit 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60)

Ohne WEG

- name = Radeon Pro Vega 64 -> keine Auswirkung
- model = Radeon Pro Vega 64 -> keine Auswirkung
- @0,model = Radeon pro Vega 64 -> Radeon Pro Vega 64 im System Profiler, nicht jedoch in Apps. Model Inject befindet sich im Framebuffer Eintrag. In GFX0 bzw. display@0 steht nach wie vor "Radeon RX Vega 64".
- Inject von PowerPlayParametern etc. funktioniert problemlos.

Mit WEG

- name = Radeon Pro Vega 64 -> Radeon Pro Vega 64 im System Profiler und in Apps.

- model = Radeon Pro Vega 64 -> Radeon Pro Vega 64 im System Profiler und in Apps.

Fazit: Egal welcher Weg, das Ergebnis ist immer das Gleiche. Nur mit WEG kann man den Namen ändern. Am Ende aber nicht mehr als eine Kuriosität da, wie auch schon mehrfach gesagt, der Name erstmal nur Kosmetik ist. Priorität hat für mich herauszufinden, welcher Mechanismus festlegt, welche Grafikkarte in VideoProc erscheint.

Beitrag von „DSM2“ vom 8. Dezember 2018, 08:16

Ah okay...

Was du machen könntest ist es die Vega ID (iMacPro) als FakeID zu speisen ala FakePCIID.

Schau dir Mal den Aufbau der Intel Kext an aus dem FakePCI Bundle.

Oder du setzt die Parameter per properties in Clover und schaust ob er dann vielleicht den Namen selbst korrekt wählt wenn er die ID kennt (iMac Pro).

Letzteres sollte sogar die einfachere Methode sein, nutze diese beim Hades Canyon und funktioniert 1A. Je nach wert ändert sich auch entsprechend der Name. Geht auch tiefer und wirkt sich auf das System aus und kratzt nicht nur an der Oberfläche, der Vega Name erscheint aber nicht in Apps... Dafür aber korrektes Renaming bzw komplette Änderung der Device ID und entsprechend andere GPU Properties. [CMMChris](#)

Beitrag von „lucifer“ vom 8. Dezember 2018, 08:20

Ich rufe mal dazwischen, ich habe gerade ein wenig Videobearbeitung gemacht und musste feststellen das alle Videoeffekte (Texte, Laufschriften, Ticker ect.) grafisch komplett verhunst sind. Hab schon einen Kollaps bekommen.

Ich musste die iGPU wieder ausschalten. Danach ist die Videobearbeitung wieder OK. Alle Effekte werden wieder richtig gerendert.

Nachteil: Die Vorschau tut wieder nicht.

So ganz fertig uns ausgereift ist das noch nicht 🤔

Beitrag von „megabyte0469“ vom 8. Dezember 2018, 08:42

Guten Morgen,

wenn ich das richtig verstanden hab hier funktioniert das ganze sowieso nur mit einer VEGA Karte. [lucifer](#): ich schätze die RX580 ist hier leider raus. Die verhält sich bei dem Thema sowieso etwas anders.

Beitrag von „DSM2“ vom 8. Dezember 2018, 09:09

Zwar noch ein alter Build meinerseits aber hab leider keine aktuelleren Bilder. Auf der Kiste läuft nach wie vor alles 1 A und es ist wie man den Bildern entnehmen kann eine 580 ohne irgendwelchen Radeon Pro Quatsch...

[megabyte0469](#)



Beitrag von „megabyte0469“ vom 8. Dezember 2018, 09:15

Danke DSM2. Ok dann falsch verstanden.

Das eine RX580 unter 10.14.2 und SMBios 18.3 korrekt läuft kann ich bestätigen.

Wie schaut es hier mit Streaming in iTunes aus ?

Das geht aber nur mit der VEGA und nicht mit der RX580. Hier muss man die Filme erst laden und dann geht das anschauen. Bei der VEGA geht das direkt ohne Download. Richtig !?!

Sorry das ich Frage aber irgendwie gibt es hier unterschiedlich Aussagen. Hatte die Frage

schon mal wo anders an die VEGA User gestellt aber nie ein Antwort erhalten.

Beitrag von „DSM2“ vom 8. Dezember 2018, 09:25

Bei der Vega hatte ich nie Probleme mit Vorschau oder Streaming oder irgendwas dergleichen. Bei dem Auftrags Build habe ich das streamen ehrlich gesagt nicht ausprobiert. Der Rechner war im Endeffekt für Videoschnitt/Bildbearbeitung und zieht jetzt immer schön fleissig an einem 10 Kerner iMacPro vorbei in Sachen Performance Video Rendering/Encoding. Und bevor es fragen gibt ja es wurde 1 zu 1 verglichen an selbem Projekt, daher kann ich die Aussage machen da selbst Bestätigt.

Beitrag von „megabyte0469“ vom 8. Dezember 2018, 09:39

@DSM

dann würdest du auf jedenfalls ein VEGA einer RX580 vorziehen bei diesem thema "streaming itunes direkt" ??

Beitrag von „DSM2“ vom 8. Dezember 2018, 10:13

Kommt drauf an wofür man sich die Karte kauft.

Generell ist die 580 keine schlechte Karte, ein Hackintosh läuft aber nun einmal nicht so wie ein PC da ist nicht einfach Karte rein und Ende ist. Wenn man keine Zeit investieren möchte ist eine Vega für den ein oder anderen eventuell die bessere Lösung... Zum laufen bringen kann man so gut wie alles man muss sich nur damit intensiv beschäftigen.

Beitrag von „CMMChris“ vom 8. Dezember 2018, 10:20

DSM2 Welche ID hat denn die Vega vom iMac Pro?

[lucifer](#) Video nach dem Rendern verhunzt? In dem Fall vermutlich falsche IGPU Konfiguration. Hatte da auch schon lustig aussehende Videos wenn ich nen Fehler in der Config hatte.

[megabyte0469](#) Nein es funktioniert nicht nur mit einer Vega. Auf die richtige Konfiguration kommt es an. Die einfachste und vermutlich in den meisten fällen funktionierende Lösung ist nach meinem jetzigen Wissensstand das: [Hardwarebeschleunigung funktioniert nicht laut Videoproc](#)

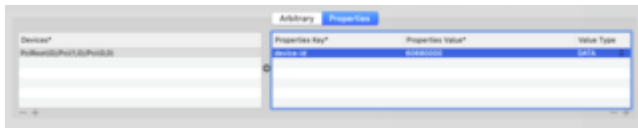
Damit läuft dann Vorschau, DRM, Safari Netflix und die DGPU wird primär fürs Encoding genutzt.

Beitrag von „DSM2“ vom 8. Dezember 2018, 16:06

[CMMChris](#)



Dann entsprechend in Clover : (Beispiel)



Beitrag von „CMMChris“ vom 8. Dezember 2018, 23:39

Tausend Dank dir. Teste ich später mal.

Edit: device-id wird leider nicht überschrieben von Clover. In IOJones ist nach wie vor die 687f zu sehen.

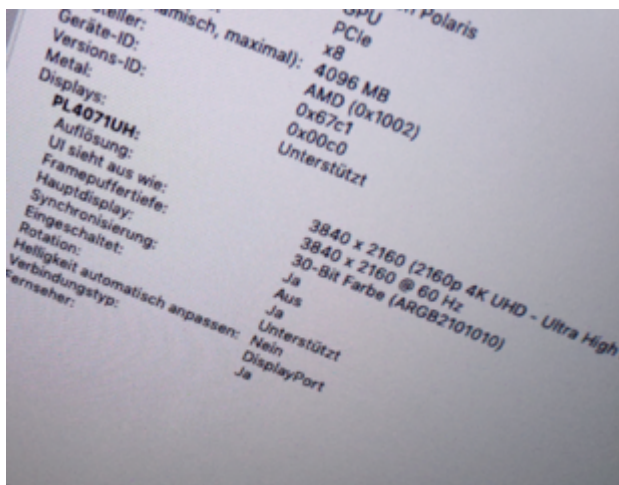
Beitrag von „DSM2“ vom 9. Dezember 2018, 03:23

Hmm... Adresse korrekt ? Mit der VegaM funktioniert das wie gesagt 1A.

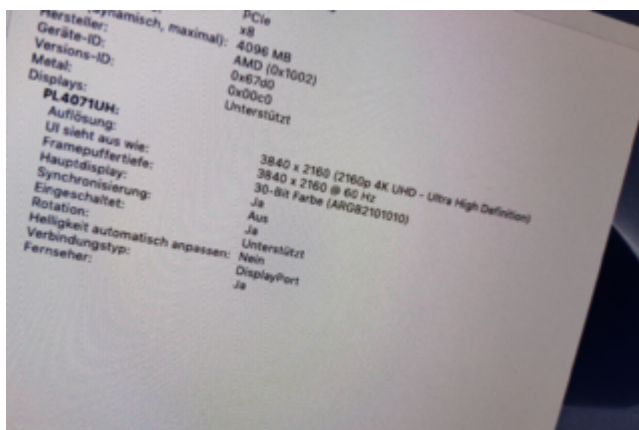
Da dein System ja aus welchen Gründen auch immer ohne WEG kein rename zulässt, mit WEG probiert?

Edit: Kleiner Beweis...

Von mir genutzte ID



Beispiel ID:



Beitrag von „megabyte0469“ vom 9. Dezember 2018, 09:08

[CMMChris](#): Danke für deine Antwort. Aber sorry ich hab nun nicht wirklich noch den Durchblick.

Hab nun zur Zeit eine Nitro+ Radeon RX Vega 64, 8192 MB HBM2 im Rechner unter 18.3 laufen.

Hier hab ich nichts an der Konfig geändert (Headless für die Igpu ist aktiv) nur die Igpu deaktiviert und nun geht alles. (ist hier im anderen Forum Eintrag bereits beschrieben)

[iTunes, Mojave und Quicksync - Erfahrungen](#)

Was muss ich nun bitte machen damit die RX580 läuft ? Du schreibst ja das das auch mit der RX580 gehen muss. Hier wird aber immer von der VEGA gesprochen und nicht der RX.

Zur Zeit schaut es so aus das alles in Videoproc auf Tot geht wenn ich die RX verwende und Igpu deaktiviere. Es gibt ja auch die Kext Treiber der auch aus der RX580 eine Readon Pro 580 macht. Braucht man das nun oder nicht. DRM und sonst geht hier dann gar nicht. Ich würde auch hier gerne Testen aber irgendwie sind die beiden Forumeinträge etwas verwirrend.

Danke für die Hilfe.

Beitrag von „CMMChris“ vom 9. Dezember 2018, 12:52

DSM2 Ja, Adresse ist korrekt. WEG ist drin und Rename über die Adresse läuft, PP_DisablePowerContainment wird angenommen, PowerPlayTable ist geladen. Die ID wird ignoriert. Habe auch schon versucht nur die ID zu injecten und den Rest rauszuwerfen, ändert nichts.

[megabyte0469](#) Es sieht wohl so aus, dass sich die RX580 auf verschiedenen Systemen unterschiedlich verhält. In anderen Foren findet man Nutzer bei denen GVA mit der RX580 ohne jegliche Kexte oder Hacks funktioniert. Es wird ja aktuell viel in die Richtung gebastelt. Früher oder später wird sich da schon eine Lösung finden. Bis dahin kannst du ja die IGPU für Video De-/Encoding nutzen.

Wenn du selbst mal was ausprobieren willst, könntest du mal versuchen deinen Hack mit dem iMacPro1,1 SMBIOS laufen zu lassen. Da würde mich sehr interessieren wie sich die Maschine dann verhält, weil in dem SMBIOS alles auf die DGPU abzielt.

Beitrag von „DSM2“ vom 9. Dezember 2018, 13:22

Komisch... Weis ehrlich gesagt auch grad nicht, woran es scheitert in deinem Fall.

Wie in meinem Beispiel weiter oben zu sehen, geht es hier problemlos.

[CMMChris](#)

Beitrag von „Doctor Plagiat“ vom 19. Januar 2019, 18:38

[CMMChris](#) Ich habe das mit dem AMD Radeon.kext jetzt auch mal probiert und die Umbenennung hat geklappt, allerdings mit WEG, da ich ohne nur im schwarzen Display lande.

Ich habe die Properties an meine RX570 angepasst:

```
model: <"Radeon Pro 570">
```

```
name: <"ATY,FlorinParent">
```

und in den Other/Kexts-Ordner gelegt.

Als Ergebnis wird mir im Systembericht nicht mehr die Radeon RX 570 sondern die Radeon Pro 570 angezeigt.

In VideoProc wird mir unter Hardwarebeschleunigung nicht mehr die UHD630 sondern die Radeon Pro 570 mit Status verfügbar in H264 und HEVC angezeigt.

DRM-iTunes-Videos laufen aber trotzdem nicht.

Beitrag von „CMMChris“ vom 19. Januar 2019, 19:05

Du änderst nur den Namen, was sich auf die Anzeige in einigen Programmen auswirkt. Die Kernfunktionalität von macOS ändert sich damit aber nicht. Die RX580 und auch ihre stärkeren und schwächeren Geschwister werden in macOS nicht für Standalone De- und Encoding unterstützt. Es ist immer das Mitlaufen der IGPU nötig und diese übernimmt auch den Hauptanteil. Schalte die IGPU ab und in Video Proc springt alles auf Rot.

Beitrag von „DSM2“ vom 19. Januar 2019, 19:45

[Zitat von Harper Lewis](#)

model: <"Radeon Pro 570">

name: <"ATY,FlorinParent">

Wo genau hast du das ausgelesen ? Hab da grad eine Idee gehabt...

EDIT:

Hat sich erledigt... Habs schon...

Beitrag von „ralf.“ vom 19. Januar 2019, 20:05

[Zitat von CMMChris](#)

Standalone [De- und Encoding](#) unterstützt.

Ich wollte nur mal nachfragen welche Programme beim [Standalone De- und Encoding](#) gemeint sind.

Final Cut Pro X auch?

Ich hab mir ne RX 460 zugelegt. Im Ryzen beim BruceX-Benchmark bin ich auf 22 Sekunden gekommen, heute waren es 23 Sekunden. Im Skylake, wenn die IGP abgeschaltet ist, 38 Sekunden.

Die Polaris rechnet das so gut wie alleine. Und in VideoProc steht bei Hardware Accel: Unavailable.

EDIT

In High Sierra

Compressor gibt es das nicht als Standalone-Programm

Beitrag von „CMMChris“ vom 20. Januar 2019, 00:41

Ich verstehe die Frage nicht. Du scheinst grundlegende Dinge nicht zu verstehen. Der Zugriff auf En-/Decoding Hardware läuft bis auf ganz wenige Ausnahmen auf OS Ebene ab. Entweder es wird unterstützt oder nicht. Bei den RX 4xx und RX 5xx Karten ist das nachweislich nicht der Fall. Ein Blick in die Plist vom Grafiktreiber verrät das, sowie der Umstand, dass Video Proc keine Hardware Encoder findet sobald die IGPU deaktiviert ist, auch wenn man Online Renderer per AppleGVA Edit oder Shiki Boot Arg erlaubt.

Weiter geht es mit BruceX. Der Test besteht nicht aus Video Dateien sondern aus Generatoren. Generatoren werden von der GPU berechnet und haben nichts mit Encoding zu tun. Natürlich lastet der BruceX Test deine AMD Karte voll aus, das die Generatoren berechnet und gerendert werden. Der Test ist nichts anderes als wenn du einen Grafik Benchmark laufen lässt. Das eigentliche Encoding läuft bei den RX 4xx und RX5xx Karten hauptsächlich auf der IGPU ab, wie man auch an der Auslastung der GPUs beim reinen Video Encoding sehen kann. Ist die IGPU deaktiviert, wird entweder auf der CPU kodiert oder es geht gar nichts mehr und Final Cut wirft zumindest beim H.264 Export mit Fehlermeldungen um sich.

Beitrag von „Doctor Plagiat“ vom 20. Januar 2019, 12:53

[CMMChris](#) Ich konnte leider nicht weiter testen, Rechner ist kaputt, hat das Deaktivieren und anschließende

Aktivieren der iGPU nicht überlebt. Das war wohl zuviel für ihn. 😄😭😬

Naja, ich hatte vorher schon Probleme, die Initialisierung der HW vom UEFI-Bios dauerte immer länger und als ich dann die iGPU wieder aktivierte blieb der Bildschirm dunkel.

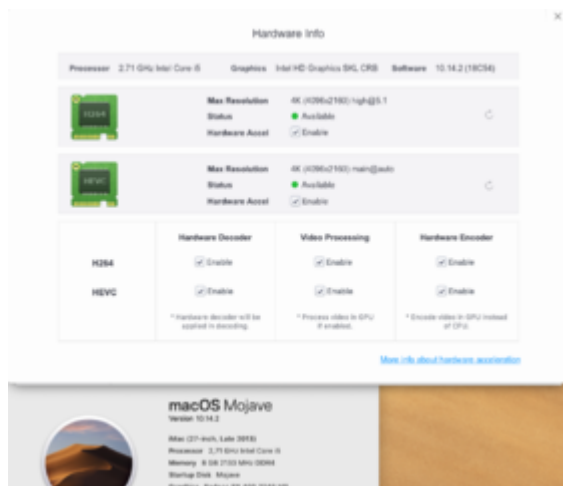
Erst nach einem Bios-Reset lief der Rechner wieder ohne Probleme, bis ich die iGPU mit ins Boot nahm. Wieder kein Bild.

Ich habe ihn jetzt in alle Einzelteile zerlegt um Board und CPU zu mindfactory zurück zuschicken. Eindeutig ein Gewährleistungsfall.

Beitrag von „ralf.“ vom 20. Januar 2019, 21:53

EDIT

Die RX 460 läuft nicht mit. Braucht vielleicht ne Vega 😊



Beitrag von „megabyte0469“ vom 21. Januar 2019, 15:05

Hallo,

bin von einer RX590 auf eine VEGA umgestiegen. Genau wegen dem Thema En-/[Decoding](#).

Kann die PowerColor Readon Vega 54 Red Dragon empfehlen.

Ultra leise Karte selbst im Spielbetrieb. einzig die Spulengeräusche die sind vorhanden aber mit abstand das leiseste was ich bis jetzt gehabt hab. Man muss schon mit dem Ohr ans gehäuse um es wirklich zu hören.

Die Spphire VEGA54 PULSE und die Sapphire VEGA54 Nitro+ waren hier um einiges lauter bei

dem Spulengeräuschen.

Hat hier jemand erfahrung mit dem Untervolting unter MacOS ?

Beitrag von „hsotnikcah1“ vom 21. Januar 2019, 16:07

[ralf.](#) laut diesem Wiki-Eintrag ist VideoProcs Anzeige nicht ausschlaggebend dafür, welche Grafikkarten angesprochen werden.

[Encoding: Herausfinden welche GPU genutzt wird.](#)

Bei mir steht ebenfalls nur die interne Intel-Graka. Teste ich jedoch wie im o.s. Link beschrieben, werden beide Grafikkarten beim Encoding aktiv.

Beitrag von „ralf.“ vom 21. Januar 2019, 18:25

@[hsotnikcah1](#)

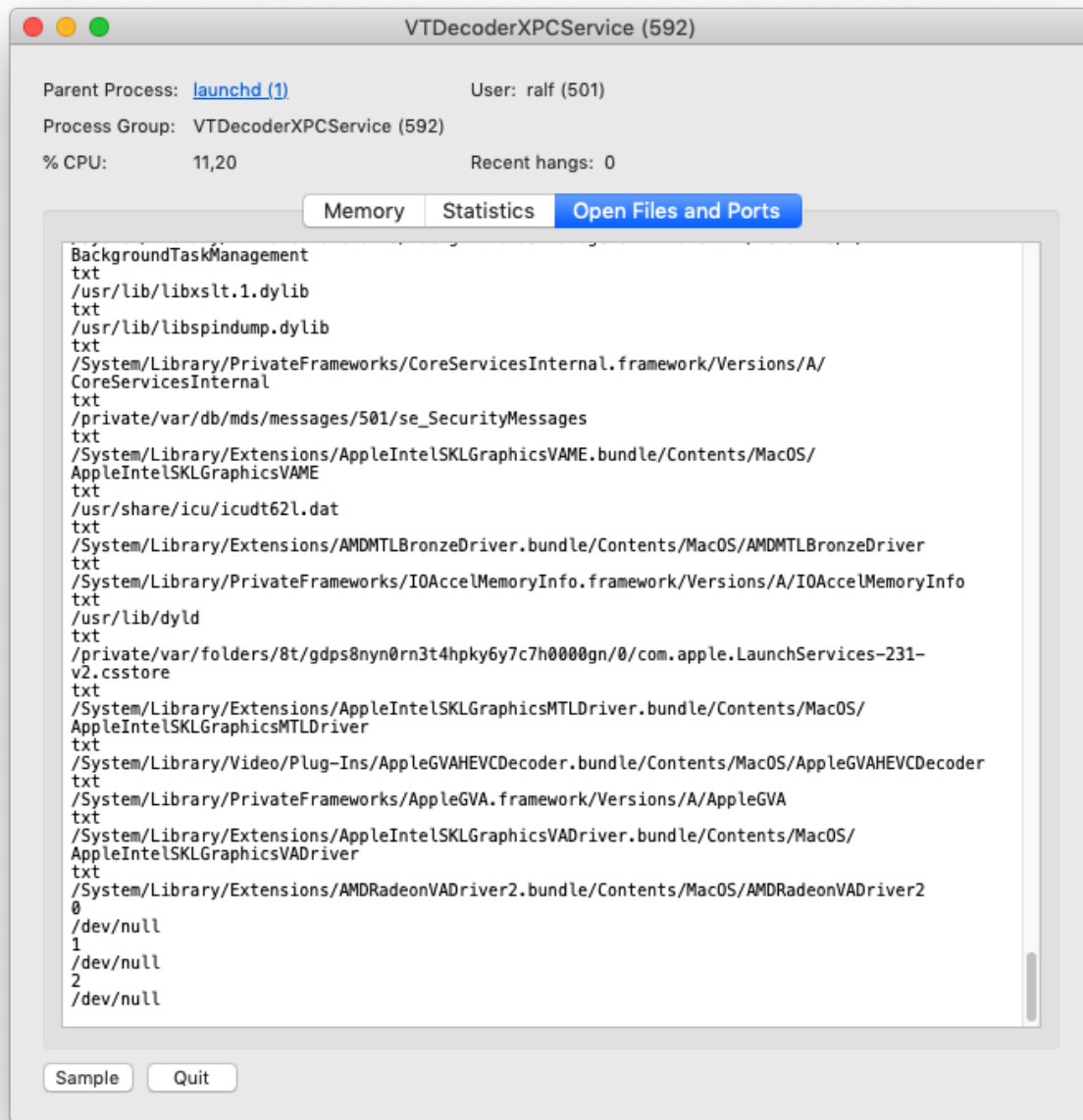
Werde ich bei gelegenheit noch mal checken. Ich hatte das Programm VideoProc ausgeführt, und ein Video teilweise transcodiert. Und dabei die Aktivitätsanzeige der Grafikkarten angeschaut. Bei der Intel ein hoher Ausschlag, bei der AMD praktisch null.

Den VTDecoder" Prozess muss ich noch mal checken.

Jedenfall gut zu wissen, dass die RX 580 dabei funzt.

EDIT

Sieht aus als Wenn die kleine Polaris doch mitarbeitet, obwohl der Aktivität Ausschlag nur minimal ist.



Beitrag von „CMMChris“ vom 21. Januar 2019, 19:14

@[hsotnik](#)cah1 Natürlich ist die Anzeige in VideoProc korrekt. Mein Wiki Eintrag ist nur eine Möglichkeit an nähere Infos zu kommen.

[ralf](#). Weder RX 4xx noch RX5xx Karten sind unter macOS für Video Encoding nutzbar und benötigen zwingen die IGPU. Verifizieren kann man das durch Prüfen der Plist vom Grafiktreiber, wo die entsprechenden Einträge fehlen, als auch durch das Deaktivieren der IGPU wonach in Video Proc alles auf rot springt und z.B. in Final Cut kein H.264 Export mehr möglich ist.

Es ist korrekt, dass bei Nutzung der IGPU (Quick Sync) auch der Video Acceleration Treiber für AMD Karten anspringt, wie jedoch an der Auslastung der GPUs zu erkennen ist, übernimmt die IGPU das Encoding. Die AMD Karte wird dabei nur für irgendwelche Nebenaufgaben genutzt. Welche genau das sind, kann ich dir nicht sagen, er springt allerdings auch bei AMD Karten an, welche gar keinen Hardware Decoder / Encoder an Bord haben.

Kurzum: Auf die reine [Encoding und Decoding](#) Leistung haben die Karten keinen Einfluss. Sie beschleunigen lediglich das Drumherum z.B. in Final Cut beim Rendern von Generatoren, Filtern, Übergängen, Effekten. Auch das lässt sich relativ leicht verifizieren. Beispiel RX580:

1. Konvertieren eines reinen(!) Videos mit Video Proc und aktiver IGPU von z.B. H.264 nach HEVC oder auch einfach nur mit demselben Codec von 4k nach Full HD.
2. Notieren der Zeit.
3. Config mit den nötigen Einstellungen für reinen IGPU Betrieb ausstatten.
4. Hackintosh abschalten, Grafikkarte ausbauen.
5. Hackintosh wieder hochfahren, checken ob die IGPU richtig läuft.
6. Test von Punkt 1 erneut ohne die AMD Karte laufen lassen. Die Zeit wird nahezu identisch sein (leichte Abweichungen im Bereich weniger Sekunden sind logischerweise normal, in einem Benchmark hat man auch nie denselben Score).

Warum ist das so?

Apple unterstützt weitestgehend nur Dinge, welche sie für ihre eigenen Produkte brauchen. Somit wird auf AMD Karten nur dann die Video [Decoding und Encoding](#) Hardware unterstützt, wenn das jeweilige Modell auch in einem Produkt zum Einsatz kommt, welches nicht über eine

IGPU verfügt. Beispiel dafür ist der iMac Pro, woher die Vega 56 / 64 Unterstützung kommt. Hier kommen Xeon Prozessoren ohne IGPU zum Einsatz, weshalb Apple in diesem Fall die AMD GPU nutzen muss.

Ist eine IGPU vorhanden, setzt Apple IMMER auf Quick Sync. Die RX580 steckt zum Beispiel im iMac 2017. Dort wird Quick Sync genutzt, weshalb es keine Unterstützung für die Encoding / Decoding Hardware der RX580 gibt.

Kann man die Unterstützung aktivieren?

Ja, kann man. Bringt aber nichts, da es nicht zuverlässig (Fehlermeldungen, Crashes, H.264 geht überhaupt nicht) funktioniert. Habe ich hier im Forum bereits mit einigen Nutzern ohne Erfolg getestet und auch im Netz findet man keine Erfolgsmeldungen.

Lohnt sich eine Vega?

Das kommt auf den Use Case an. Wer viele und große Video Projekte rendern muss und viel Material konvertiert der profitiert natürlich von einer Vega. Zeit ist Geld. Wer mit Video Proc experimentiert, wird über die IGPU in aktuellen Intel Chips selten mehr als 80fps sehen. Mit den Vega Karten sind bei reiner Nutzung der Vega ohne Quick Sync (IGPU) je nach Format und Auflösung Werte jenseits der 200fps möglich.

Hoffnung dank eGPU Unterstützung?

Hoffnung, dass sich an der Situation in der Zukunft etwas ändert, darf man haben. Im Zuge der eGPU Unterstützung wird Apple künftig stärkeres Augenmerk auf die Unterstützung diverser Grafikkarten legen. Noch befindet sich das Thema eGPU in den Kinderschuhen. In Zukunft könnte Apple an dem Vorgehen durchaus etwas ändern und eGPUs generell fürs [Encoding und Decoding](#) nutzbar machen. Damit würde dann auch breite Unterstützung auf den Hacks Einzug halten.

Beitrag von „megabyte0469“ vom 22. Januar 2019, 08:18

[CMMChris](#) geht das tool mit einer Vega54 überhaupt ?

Lese immer nur Vega64 bei den Beispielen

Beitrag von „DSM2“ vom 22. Januar 2019, 08:29

Du meinst wohl Vega 56! Eine Vega 54 gibt es jedenfalls nicht.

Probier es doch einfach aus, viel kann nicht passieren!

Falls Kext unbrauchbar und Probleme macht, einfach Kext laden verhindern per Clover.

Beitrag von „megabyte0469“ vom 22. Januar 2019, 09:01



Stimmt es soll natürlich VEGA56 heißen.

Beitrag von „CMMChris“ vom 22. Januar 2019, 10:55

Du kannst auch VGTab-en nutzen, das hat einen eigenen Modus für die Vega 56.

Beitrag von „megabyte0469“ vom 22. Januar 2019, 11:45

Ok hab ich im Netz schon gefunden. Was muss da dann in clover rein ? Nur der kext driver der generiert wird ?

Beitrag von „DSM2“ vom 22. Januar 2019, 12:05

Kext in EFI unter EFI/CLOVER/kexts/other rein und reboot 😊

Beitrag von „Temporaer“ vom 24. Mai 2020, 11:33

Zitat von griven

Nun sie werkeln ja auch harmonisch zusammen redbelt wenn eben die Voraussetzungen dafür geschaffen wurden. Es ist eben ein Trugschluss zu denken das SMBIOS hätte nur was mit der Optik zu tun bzw. war das ganz früher mal so aber in der Zwischenzeit hat sich einiges getan 😊

Mal ein konkretes Beispiel:

Ausgehend von Deiner CPU würde man wohl als Modell am ehesten den iMac 18.3 wählen wollen denn das ist ein ziemlich perfektes Abbild von Deinem Setup. Die CPU ist identisch (i7-7700K) und auch die Grafik ist nahezu identisch RX580 vs. Radeon Pro 580 besser geht es SMBIOS technisch gar nicht. Gucken wir dem Bruder Deines Systems mal unter Die Haube bzw. hinter das Display. Wir haben also einen iMac mit der BoardID Mac-BE088AF8C5EB4FA2 das ist gut zu wissen denn damit bewaffnet können wir mal in der AppleGraphicsDevicePolicy gucken was der so können sollte. Der iMac 18.3 wird mit der config3 gelistet und die wiederum besagt folgendes:

Code

1. <key>Config3</key>
2. <dict>
3. <key>GFX0</key>
4. <dict>
5. <key>EDID</key>
6. <dict>
7. <key>index</key>
8. <integer>0</integer>
9. </dict>

10. <key>FeatureControl</key>
11. <integer>12</integer>
12. <key>unload</key>
13. <false/>
14. </dict>
15. <key>IGPU</key>
16. <dict>
17. <key>unload</key>
18. <true/>
19. </dict>
20. </dict>

Alles anzeigen

Wir wissen nun das Gerät hat 2 Grafikeinheiten einmal die dGPU und die iGPU ferner wissen wir Apple erwartet das die iGPU im ACPI als solche zu finden ist und es erwartet wird das die dGPU als GFX0 im System vorhanden ist. Nun ist es bei nicht Apple Geräten oft so das die iGPU als GFX0 definiert ist im ACPI und die dGPU sich unter PEGP oder PEG verbirgt. Genau wissen wir das in Deinem Fall aber erst wenn wir einen Blick in die DSDT gewagt haben. Wichtig ist jedenfalls schon mal das wir diese, von Apple verlangte, Nomenklatur einhalten ansonsten führt das unweigerlich zu Chaos.

Was sagt uns unser Blick in den Kext noch nun er sagt uns das für die iGPU der Key `unload` auf `true` gesetzt wird und das meint nichts anderes als das der Framebuffer für die iGPU nicht geladen bzw. wieder entladen werden soll. So weit so gut denn den Framebuffer (Bildschirmtreiber für die Darstellung von 2D Inhalten auf Bildschirmen) brauchen wir ja auch gar nicht denn das übernimmt ja sowohl im iMac als auch in Deinem Fall die AMD Karte. Unser erstes Arbeitspaket lässt sich also hieraus ableiten:

1. In die DSDT gucken und herausfinden wie die Grafikkarten im ACPI angesprochen werden.
2. Dafür sorgen das dies macOS konform geschieht entweder über Clover oder direktem DSDT Patch.
 - 2a. Wenn über Clover dann darauf achten das man als erstes GFX0 in iGPU umbenennt und erst dann zum Beispiel PEGP in GFX0.
3. Mittels Clover oder DSDT einen Connectorless PlattformID injecten (0x59120003).

4. Dafür Sorge tragen das es ein DEVICE IMEI gibt entweder durch umbenennen des vorhandenen MEI Devices oder durch einfügen eines Fake Devices.

Stellt sich jetzt natürlich die Frage warum der ganze Driss mit dem Apple konformen umbenennen im ACPI und die Antwort darauf ist relativ einfach! MacOS ist nicht Windows und hat für alles und jeden einen Treiber und es ist auch nicht Linux das sich für jede Vendor und DeviceID schon was passendes aussucht sondern macOS erwartet einen Mac und geht daher davon aus die nötigen Informationen zum einen aus dem SMBIOS zu ziehen zum anderen aber auch aus dem ACPI und eben so sicher wie es davon ausgeht dort die entsprechenden Informationen zu finden macht es davon auch abhängig welche Funktionen auf der erkannten Hardware bereit gestellt werden (AirPlay, H264 en- und decoding, HVEC en- und decoding usw.)

Nachdem wir das erledigt haben sind wir dem Ziel schon ein gutes Stück näher gekommen denn unser nicht ganz echter iMac kennt nur bestenfalls seine iGPU und auch eine (bewusst nicht seine) dGPU und kann zumindest schon mal mit den Informationen über die iGPU arbeiten. Warum schreibe ich eine und nicht seine dGPU auch das ist wieder relativ leicht zu erklären denn der Apple verwendet ja nichts von der Stange sondern die Polaris Karten in den Mac's sind nicht irgendwelche Polaris Karten (haha) sondern es sind Radeon Pro Karten. Der Fachmann staunt und der Laie wundert sich was macht nun eine Brot und Butter Polaris Karte zu einer Radeon Pro und was zum Teufel ist Radeon Pro überhaupt?

Hier kommt wieder der von [CMMChris](#) bereits erwähnte Fakt ins Spiel. Apple ist auf die glorreiche Idee gekommen so genannte "egpu developer kits" zu verkaufen in denen eine AMD RX verbaut war und die es Entwicklern ermöglichen sollte schon mal an der Leistungsfähigkeit der Metal API sowie der kommenden dGPU Generation zu naschen ohne dafür konkret einen Mac in Petto zu haben der das bereits verbaut hatte. Jetzt hat man also eine ganze Horde von Entwicklern die sich die Grafikkiste gekauft haben und wie überzeugt man die nun davon das Dingen wieder in die Ecke zu stellen und einen hochpreisigen Mac zu kaufen ach eigentlich ganz einfach man sparrt sich ein paar wenige, aber essentielle, Funktionen auf die sich nicht über Thunderbolt tunneln lassen, lötet die gleiche GPU auf das LogicBoard verpasste dem Dingen aber vorsichtshalber vorher eine andere DeviceID und zack hat man eine Radeon Pro ersonnen. Der Fachmann schüttelt den Kopf und kauft trotzdem denn gibt ja keine Alternative und der Laie wundert sich schon wieder und huldigt Apple oder so ähnlich...

Was sagt uns das nun? Die RX580 ist genau so Pro wie die Radeon Pro 580 man muss macOS das nur verklickern und das geht indem man der die RX entsprechend verkleidet zum Beispiel hiermit: [AMDRadeonPro.kext.zip](#) wobei es sich hierbei um einen alten Bekannten handelt nämlich um den PropertyInjector.kext von [Brumbaer](#). Natürlich kannst Du das Ding jetzt nicht einfach reinwerfen und hoffen das es geht vielmehr musst Du in der info.plist des Kexts die Device und VendorID unter PCIPrimaryMatch noch an Deine Gegebenheiten anpassen aber ist das einmal getan steht dem fast echten iMac 18.3 nichts mehr im Wege.

Puh nun ist es mehr Text geworden als ich eigentlich schreiben wollte aber vielleicht hilft das ein wenig Licht ins Dunkle zu bringen...

Alles anzeigen

Ich bin kein MAC-Nutzer. Diesen Beitrag fand ich dennoch hilfreich und habe mich heute hier registriert, um mich für diesen Beitrag zu bedanken. Wirklich, sehr gut gemacht, [griven](#)

Beitrag von „griven“ vom 24. Mai 2020, 11:34

Na da kann ich nur danke für das Danke sagen 😊