

Erledigt

Installation von Mojave bricht zwei Minuten vor dem Ende ab...

Beitrag von „Godomac“ vom 9. November 2018, 16:21

Tja, wie fange ich an... Also es ist mein erster Versuch, einen Hackintosh ans Laufen zu bekommen.

Leider bricht die Installation von Mojave immer 2 Minuten vor dem Ende ab. Das System friert sozusagen ein.

Ich hänge hier mal meine config.plist und das gezippte EFI an.

[EFI.zip](#)

[config.plist.zip](#)

Vielleicht mag mir ja jemand helfen - bin dankbar für jeden Tipp oder Hinweis!

Viele Grüße

Guido

Beitrag von „FARV“ vom 9. November 2018, 17:15

Ich hatte schon versucht, dem Guido live per Videokonferenz zu helfen, doch leider bin ich auch mit meinem Latein am Ende. Vermutlich ist es nur eine Kleinigkeit, die da noch quer schießt.

Wäre toll, wenn sich ein Spezi mal die von mir erstellte und von Guido gepostete EFI/config.plist anschauen würde.

In meinem System habe ich leider "nur" eine RX580 und keine Vega 64, so dass ich dazu nicht viel beitragen kann.

Muss er evtl. die iGPU im Bios deaktivieren oder spezielle Einstellungen vornehmen?

Das Bios von seinem MB ist mit dem von meinem Asus Prime Z370-A auf dem ersten Blick nahezu identisch (von der Bedienung her). Und soweit ich es beurteilen kann, hat er eigentlich alles richtig eingestellt.

Doch letztendlich hängt der Rechner bei der Installation von Mojave dann, wenn die Fortschrittsbalken "noch 2 Minuten" anzeigt.

Ach ja... Der Bootstick wurde nach Forumsanleitung mit TINU und Clover erstellt. Kexte sind über CCC installiert worden. Der Stick steckt in einem USB-2 Port und wird darüber gebootet.

Beitrag von „kuckkuck“ vom 9. November 2018, 17:55

[FARV](#) hat da sicherlich schon einiges probiert und die Hardware ist ähnlich.

Um zu verhindern, dass wir hier umsonst rumprobieren, hast du mal die Installationsdatei komplett frisch und neu aus dem AppStore geladen und einen Stick per Terminal erstellt? Nicht, dass irgendwelche Installressourcen defekt sind, der Stick startet schließlich bis ins Installmenü.

Habt ihr es auch mal mit einer nicht so stark modifizierten config probiert?

Beitrag von „FARV“ vom 9. November 2018, 18:11

Den Bootstick an sich habe ich nicht überprüft. Wäre ne Idee, den noch einmal neu zu erstellen.

Die Modifikationen an der config.plist stammen von mir und lehnen sich an meine eigne config.plist an.

Zuvor war die "default" config.plist, die Clover selbst auf den Stick schreibt im Einsatz.

Beitrag von „Godomac“ vom 9. November 2018, 19:46

Ich lade gerade Mojave neu herunter... Werde mal einen neuen Stick erstellen. Zumindest einen Versuch wert 😊

Beitrag von „FARV“ vom 10. November 2018, 02:33

Ich habe überlegt, ob man evtl. nicht die iGPU ausschaltet und den Rechner per CC zum "iMac pro 1,1" deklariert.

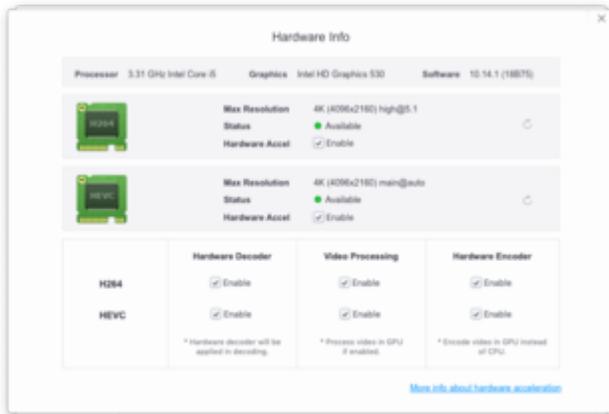
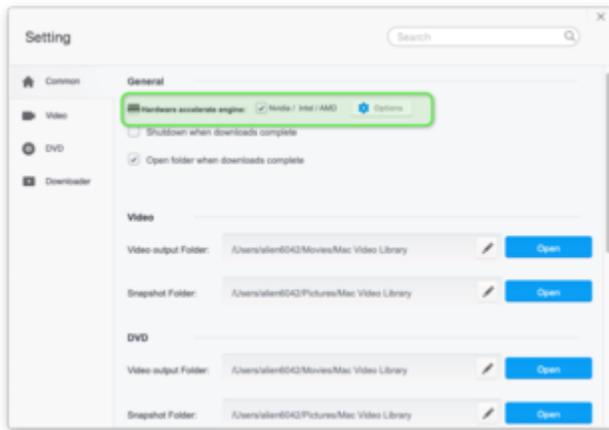
Wäre das nicht einen Versuch wert?

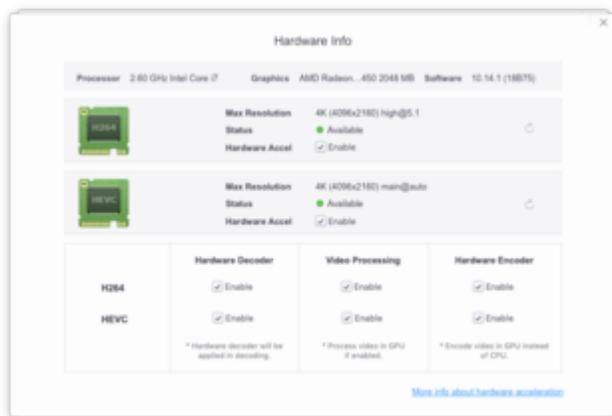
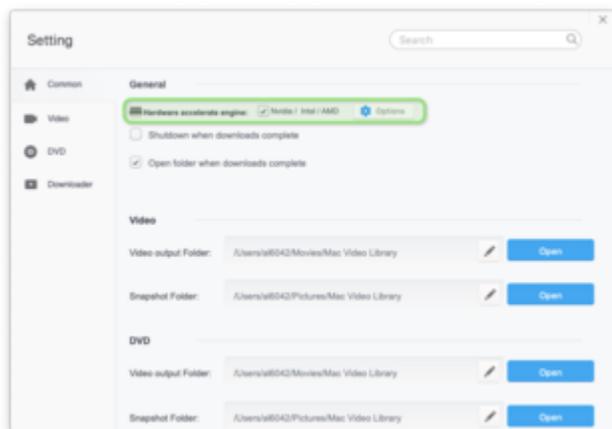
Schließlich ist ja ne Vega eingebaut, die Apple bisher ja auch nur im iMac pro einsetzt?!

Beitrag von „al6042“ vom 10. November 2018, 12:23

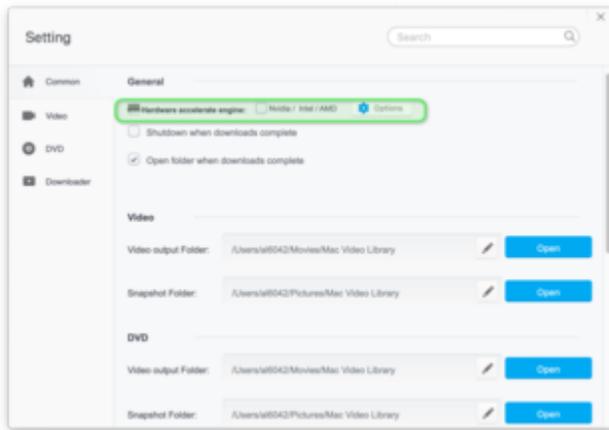
Ich habe das eben mal geprüft und verglichen.

Das Problem scheint an den Einstellungen zu liegen, da bei meinen Skylake, mit Single HD530, und dem MacbookPro, mit HD530+Radeon Pro450, die Unterstützung funktioniert:

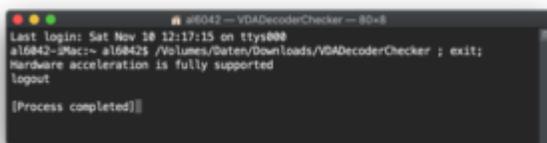




Während bei meinem Kabylake-System, mit HD630 (connectorless)+Vega 56, das seit 10.14.1 nicht mehr funktioniert:



Bei mir steht in den "Custom Flags" noch der "shikigva=1", der natürlich zu folgendem Ergebnis beim VDADecoderCheck führt:



Während der Skylake und das MacbookPro wie folgt darauf reagieren:

```
al6042 — VDADecoderChecker — 80x12
Last login: Wed Nov 7 15:56:49 on console
al6042-iMac17:~ al6042$ /Volumes/Daten/Downloads/VDADecoderChecker ; exit;
Hardware acceleration is fully supported
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...3 completed.

[Process completed]]
```

```
al6042 — VDADecoderChecker — 80x13
Last login: Sat Nov 10 12:47:08 on ttys000
al6042-MBP13-2:~ al6042$ /Users/al6042/Downloads/VDADecoderChecker ; exit;
Hardware acceleration is fully supported
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...22 completed.

[Process completed]]
```

Beitrag von „CMMChris“ vom 10. November 2018, 12:41

[al6042](#) Bei mir ist das Vega Encoding bei der 64er auch kaputt seit 10.14.1.

Beitrag von „al6042“ vom 10. November 2018, 13:09

[CMMChris](#)

Du arbeitest bekanntlich ohne WEG, der bei mir noch integriert ist, um die Shiki-Aktionen nutzen zu können.

Ich habe bisher mal ohne "shikigva" und mit verschiedenen Werten aus [Noir0SX Liste](#) getestet, aber keine Verbesserung feststellen können.

Das führt mich zu Test mit den älteren WEG Version (1.2.2 & 1.2.3) mit und ohne "shikigva=1", was aber zum gleichen Resultat führte.

Ich glaube, dass da an den AMD Kexten von Apple fundamental geschraubt wurde.

Beitrag von „CMMChris“ vom 10. November 2018, 13:55

WEG ändert nichts bei mir. Habe ja viel probiert bei mir, sie meinen Beitrag dazu.

Beitrag von „FARV“ vom 10. November 2018, 13:59



So, wir haben jetzt High Sierra auf dem Rechner installieren können.

Damit gibt es nun einen Hackintosh-Nutzer mehr.

Als config.plist habe ich als Basis für den Rechner die Clover-CFL plist von [al6042](#) genutzt.

Ton läuft, Netzwerk läuft, iCloud läuft.... Guido richtet sich das Teil jetzt erst einmal ein. Update auf Mojave erfolgt dann später.

Beitrag von „al6042“ vom 10. November 2018, 14:07

[CMMChris](#)

Schon klar... 😊

Mir ging es eher darum auch zu klären, dass der WEG bisher unter 10.14.1 nicht wirklich helfen kann.

Wie gesagt, scheint es, dass die grundlegenden AMD-Kexte von Apple als Ursprung der aufgefallenen Änderung in der H.264 Unterstützung bei "connectorless" eingebundenen iGPUsm in Kombi mit AMD-Karten eine Rolle spielen.

Daher hast du den gleichen Effekt, auch ohne Nutzung des WEGs.

Beitrag von „CMMChris“ vom 10. November 2018, 14:14

Hast du auf einem Hackintosh mit Quick Sync in Video Proc eigentlich schonmal die DGPU in der Anzeige gesehen? Auf den echten Macs wird mit Quick Sync dort (zumindest bei meinen) die DGPU angezeigt. Auf den Hackintosh dagegen sieht man immer die IGPU. Würde mich mal interessieren woran das liegt.

Beitrag von „al6042“ vom 10. November 2018, 14:19

In meinen obigen Screenshots zeigt mein MBP ebenso die Radeon 450 Pro im VideoProc an.

Diese wird auch nicht "connectorless" bedient, da sie ja tatsächlich auch den internen Display betreibt:



Theoretisch müsste das auch bei einem Hackintosh möglich sein, ohne die IGPU "Connectorless" zu injecten.

Ich bin aber nicht der Programmier-Fuchs, um den entsprechenden Part in den Treibern zu finden... 😊

Beitrag von „kuckkuck“ vom 10. November 2018, 14:35

[Zitat von al6042](#)

zeigt mein MBP ebenso die Radeon 450 Pro im VideoProc an.

Diese wird auch nicht "connectorless" bedient

Meinst du jetzt die AMD450 oder die HD530? Oder wieso sollte eine AMD450 über den Framebuffer connectorless sein?

Beitrag von „al6042“ vom 10. November 2018, 14:43

Sorry... war missverständlich.

Ich meinte natürlich, dass die HD530 im MBP nicht als "connectorless" betrieben wird, was ja auch der Screenshot zeigt.

Beitrag von „kuckkuck“ vom 10. November 2018, 14:45

Bist du dir sicher (mal mit IOReg oder DPCI Manager abgecheckt?) weil manchmal werden die IGPU's in Mojave wohl trotz connectorless im Systembericht angezeigt...

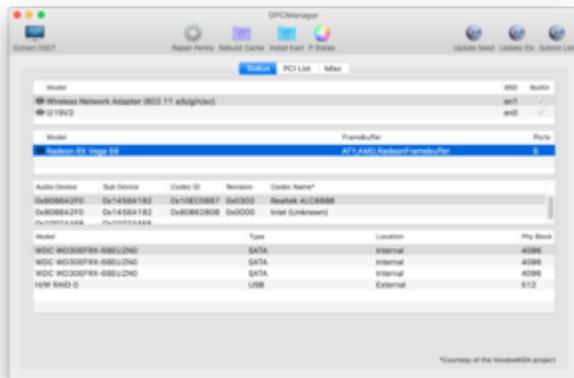
Beitrag von „al6042“ vom 10. November 2018, 14:58

Ich verstehe die Frage gerade nicht...

Ich injecte die ig-platform-id 0x59120003 per DSDT, diese wird auch im IOReg wie folgt dargestellt:

```
AAPL,ig-platform-id      Data      <03 00 12 59>
```

Im DPCIManager taucht die HD630 und "PCI List" auf, aber weder in dessen "Status"-Fenster, noch in den Systeminformationen:



Hier kannst du auch sehen, das trotzdem HDMI-Audio darüber möglich wäre, wenn auch kein Bild ausgegeben würde.

Beitrag von „kuckkuck“ vom 10. November 2018, 15:01

Sorry, die Frage bezieht sich auf das MBP. Bist du dir sicher, dass die HD530 auf dem MBP nicht doch connectorless ist? Hast du das mal per IOReg auf dem MBP überprüft?

Beitrag von „al6042“ vom 10. November 2018, 15:06

Äh... der obige Screenshot zeigt, dass über die IGPU der interne Display angebunden ist, was mit folgenden Angaben unterstützt wird:



Dazu noch die Ziele aus dem IOReg, welche den gleichen ig-platform-id-Wert ausgibt:

IOPropertyKey	Number	UXU
AAPL,ig-platform-id	Data	<06 00 1b 19>
IOName	String	dien1au

Beitrag von „kuckkuck“ vom 10. November 2018, 15:11

Interessant, bist du dir auch sicher, dass Decoding (und die Anzeige in VideoProc) sich auf die HD530 bezieht und nicht die AMD450 (kann die nicht auch HW Decoding?).

Weil ansonsten hat Apple was an den Treibern verändert und dem Treiber VirtualScreenChange etc beigebracht...

Beitrag von „CMMChris“ vom 10. November 2018, 15:11

Das Problem ist, dass Video Proc die Vega offenbar überhaupt nicht erkennt. Ob die iGPU nun Connectorless ist oder nicht spielt bei echten Macs keine Rolle. In Video Proc wird immer die DGPU falls vorhanden angezeigt. Bei einem Hackintosh nicht (schon immer, nicht erst seit 10.14.1). Bei meinen iMacs ist die iGPU Connectorless eingebunden, sprich kein Framebuffer zugewiesen, nur der Accelerator ist aktiv.

Wenn ich die IORegs von meinen iMacs mit meinem Hackintosh vergleiche sieht da bei iGPU und DGPU alles identisch aus - bis auf fehlendes AGPM bei der DGPU (mit WEG wird auch AGPM geladen). Und trotzdem zeigt Video Proc die Vega nicht an wenn Quick Sync aktiv ist, sondern die iGPU.

Vor 10.14.1 hat Video Proc die Vega angezeigt wenn man die iGPU deaktiviert hat. Seit 10.14.1 steht in Video Proc dann bei Grafikkarte nur noch "N/A" und Encoding funktioniert in keiner App mehr korrekt (H.264 Rot, HEVC grün, bei HEVC wird aber trotzdem hauptsächlich die CPU belastet und die Vega nur halbherzig).

Beitrag von „kuckkuck“ vom 10. November 2018, 15:17

Zitat von CMMChris

Ob die iGPU nun Connectorless ist oder nicht spielt bei echten Macs keine Rolle.

Vielleicht im Bezug auf VideoProc, aber das spielt eindeutig eine Rolle, denn ist der iGPU ein Port zugewiesen und der HWDecoder der iGPU wird daraufhin genutzt, muss der Treiber einen VirtualScreenChange hin und wieder zurück machen (bzw wäre es sein natürliches Verhalten) und das ganze muss der Treiber können, oder eben auch nicht. Und bisher konnte das der Treiber (gut ich habe nicht jegliche iGPU+ded.GPU Kombis der Welt probiert) nicht, ein erzwingener VirtualScreenChange bei einem Hackintosh (mit nicht connectorless iGPU) führte zu Programmcrashes, System crashes oder freezes.

Sollten aber inzwischen Macs iGPUs mit Framebuffern haben, die Ports besitzen und sowohl den HW Decoder der ded. GPU als auch der iGPU nutzen können, bedeutet das, dass der Treiber inzwischen VirtualScreenChange beherrscht, was die Notwendigkeit von connectorless

iGPUs im Prinzip verschwinden lässt.



Sorry für den komplizierten Text, ich hoffe es verstehen nicht alle nur

Beitrag von „al6042“ vom 10. November 2018, 15:20

Dann sollte ich ggf. mal mit normaler id-platform-id starten und das checken, oder?

Beitrag von „kuckkuck“ vom 10. November 2018, 15:21

Kannst du machen, interessanter wäre aber eher noch zu wissen, ob das MBP wirklich sowohl den HW Decoder der iGPU als auch den der ded. GPU nutzen kann, obwohl beide einen Framebuffer mit zugewiesenen Ports benutzen. Hat die AMD Pro 430 überhaupt einen HW Decoder?

Beitrag von „CMMChris“ vom 10. November 2018, 15:35

Also bei mir crasht der Hackintosh in FCPX wenn die iGPU nicht connectorless ist (instant reboot).

Beitrag von „al6042“ vom 10. November 2018, 15:37

Test war wenig erfolgreich:



Laut der folgenden Seite sollte die VCE3.0 als HW-Decoding für die Pro450 greifen -> https://en.wikipedia.org/wiki/Video_Coding_Engine

Beitrag von „kuckkuck“ vom 10. November 2018, 15:38

[CMMChris](#) Und der Bericht zeigt sowas in der Art?

Code

1. [FFHGRendererInfo initWithLocation:] CGLSetVirtualScreen failed - didn't expect to get here. Investigate why this code ran: [FFHGRendererInfo initWithLocation:] CGLSetVirtualScreen failed

Beitrag von „CMMChris“ vom 10. November 2018, 16:29

In welchem Szenario? Connected IGPU? Nein, Beschleunigung gibt dabei grünes Licht. In Video Proc läuft Quick Sync damit stabil. Nur FCPX crasht.

Beitrag von „kuckkuck“ vom 10. November 2018, 16:46

Und steht sowas im FCPX Crash report?

Beitrag von „CMMChris“ vom 10. November 2018, 16:59

Es gibt keinen FCPX Crash report. Wie gesagt gibt es einen Instant Reboot (Kernel Panic). Den Report darüber habe ich mir aber nie angesehen weil es mich nicht interessiert hat, Connectorless läuft es ja. Ich kann das aber gerne fix triggern und den Report anhängen wenn gewünscht.

Beitrag von „kuckkuck“ vom 10. November 2018, 17:11

Wäre an der Stelle ganz interessant, wenn das geht – gerne 😊

Beitrag von „FARV“ vom 10. November 2018, 17:14

Leute, Leute, da hab ich ja was losgetreten. 😊

Aber zum Thema...

Bei den MacBooks (pro) kann man ja in den Energie-Einstellungen zwischen "Höherer Performance" und "Energie sparen" hin und her schalten.

macOS nutzt dann die DGPU oder die iGPU, je nach Anwendungsfall.

Bei den iMacs ist die iGPU idR. ja connectorless eingebunden. Da gibt es dieses Umschalten, wie bei den MacBooks nicht.

Ich vermute, dass das was mit dem SMBios-Einstellungen zu tun hat. Sprich, dass es einen Unterschied macht, ob Rechner als MacBook (pro) oder iMac von macOS identifiziert wird.

Doch das müsste man ggf. mal austesten.

Beitrag von „CMMChris“ vom 10. November 2018, 17:14

Fehlanzeige, wird vom System offenbar nicht als Kernel Panic erkannt. Es erscheint weder beim Start die Meldung "System was automatically rebooted after panic" noch der Report beim Systemstart. Der Rechner geht einfach nur aus und startet neu. Hab schon die Konsole durchforstet aber finde nix aussagekräftiges. 😞

Beitrag von „Na7ets“ vom 14. November 2018, 15:09

[FARV](#) Könntest du mir sagen was jetzt genau geändert wurde? Das bei Godomac die installation komplett funktionierte. Habe nämlich gerade das selbe problem. Bei noch "ungefähr 2 Minuten" friert die Installation ein.