

# Cube i7 Book Tablet Skylake Core M3-6y30

Beitrag von „strega“ vom 19. Januar 2017, 22:42

Wäre schon schön, aber ich hab weiß nicht mal an welcher Stelle er sich aufhängt und dieses "Null object" findet.

EDIT: Ok ich habs hinbekommen mit einem Trick:

Wir wussten ja, dass alle SSDTs untereinander ohne Fehler decompilen. (also ohne DSDT)

Also hab ich einzeln immer versucht eine SSDT nach der anderen in den Ordner reinzuschieben und dann immer wieder alle \*.aml decompiled. Tatsächlich liegt der Fehler in SSDT-3.aml, die durch fehlende Referenzen (und dadurch falsche Argumentanzahl bei Methoden, etc.) so schwerwiegend falsch decompiled, dass iasl abbrechen muss.

Also hab ich Folgendes gemacht:

1. Alle SSDTs untereinander decompiled.
2. Die entstandene SSDT-3.dsl hab ich versucht zu compilieren. => VIELE FEHLER
3. Dann hab ich die ref.txt um die Parameter erweitert, die er nicht kannte und die dadurch Fehler verursachen. Witziger Weise waren die fast alle in DDST definiert, deshalb konnte ich guten Gewissens die Parameteranzahl übernehmen. Dann wieder alle decompiled um eine neue SSDT-3.dsl zu erhalten, welche mehr Referenzen enthält.
4. Wieder versucht SSDT-3.dsl zu compilieren => KEINE FEHLER MEHR.
5. Die "gefixte" SSDT-3.aml nun wieder zu den anderen Ausgangs .aml files gepackt und die DSDT, welche ja bisher außen vor war, mit in den Ordner.
6. Alle \*.aml decompiled => Durchgelaufen!

Und ja letztendlich musste ich SSDT-3 jetzt "opfern", damit die anderen wenigstens halbwegs richtig sind, aber letztendlich fehlen immer irgendwelche Verweise, sodass das idiotisch ist zu versuchen ALLE SSDTs zu fixen. Wenn wir SSDT-3 später brauchen sollten, kann die zur Not auch von Hand gefixt werden (die meisten Verweise kommen ja aus der DSDT). Wenn wir die aber eh nicht patchen müssen, dann ist es auch völlig egal, dass wir sie nicht so korrekt wie die anderen decompiled haben.

EDIT2: Achso ich hab mal die ref.txt angehängt, mit der sich SSDT-3 decompilieren und dann wieder compilieren lässt 😊