

Sechs Paare von 16 Ziffern, die oft mit der Mac-Adresse der Netzwerkkarte übereinstimmen. Es gibt jedoch, Meldungen, dass der Dienst mit beliebigen Ziffern arbeitet. Ab der Revision 3051 ist es möglich, daß sie schreiben `<string>UseMacAddr0</string>` und Clover wird die Mac-Adresse Ihrer Netzwerk-Karte ermitteln. Das Verfahren funktioniert nicht bei jedem, also prüfen sie es. Und das Wichtigste: Wenn Sie sich für iMessage anmelden, handelt es sich um einen kostenpflichtigen Dienst, sie müssen Ihre echtes Bankkonto-Karte angeben, von der Sie mit einem Dollar belastet werden. Wer versucht es umsonst zu nutzen, erhält die Meldungen wie "Rufen Sie Apple an".

Im Jahr 2015 kam ElCapitan 10.11 mit neuen Sicherheitsanforderungen heraus. SIP = System Integrity Protection (Schutz der Systemintegrität). Standardmäßig ist der Schutz aktiviert und erlaubt es Ihnen nicht, Ihre Kexte und ihre eigenen Systemdienstprogramme zu installieren. Um sie zu deaktivieren, gibt es in Clover die Möglichkeit, neue Parameter im NVRAM zu setzen

```
<key>CsrActiveConfig</key>
<string>0x3E7</string>
```

```
<key>BooterConfig</key>
<string>0x28</string>
```

Dies sind Bitmasken mit möglichen Bitwerten

```
/* Wurzellose Konfigurationsflags */
#define CSR_ALLOW_UNTRUSTED_KEXTS    (1 << 0)
#define CSR_ALLOW_UNRESTRICTED_FS    (1 << 1)
#define CSR_ALLOW_TASK_FOR_PID       (1 << 2)
#define CSR_ALLOW_KERNEL_DEBUGGER    (1 << 3)
#define CSR_ALLOW_APPLE_INTERNAL     (1 << 4)
#define CSR_ALLOW_DESTRUCTIVE_DTRACE (1 << 5) /* name      deprecated */
#define CSR_ALLOW_UNRESTRICTED_DTRACE (1 << 5)
#define CSR_ALLOW_UNRESTRICTED_NVRAM (1 << 6)
#define CSR_ALLOW_DEVICE_CONFIGURATION (1 << 7)
#define CSR_ALLOW_ANY_RECOVERY_OS    (1 << 8)
#define CSR_ALLOW_UNAPPROVED_KEXTS   (1 << 9)
#define CSR_ALLOW_EXECUTABLE_POLICY_OVERRIDE (1 << 10)
```

```
/* Bitfelder für boot_args->flags */
#define kBootArgsFlagRebootOnPanic   (1 << 0)
#define kBootArgsFlagHiDPI           (1 << 1)
#define kBootArgsFlagBlack            (1 << 2)
#define kBootArgsFlagCSRActiveConfig (1 << 3)
#define kBootArgsFlagCSRPendingConfig (1 << 4)
#define kBootArgsFlagCSRBoot         (1 << 5)
#define kBootArgsFlagBlackBg         (1 << 6)
#define kBootArgsFlagLoginUI         (1 << 7)
```

Die Standardwerte entsprechen der Deaktivierung des Schutzes. Wenn sie aufgrund einer gewissen Paranoia den Schutz aktivieren wollen, setzen sie ihn auf 0 (Null).

Die folgenden drei Parameter sind ausgenommen, da sie vom System aus der Steuerung eingestellt werden müssen Clover-Panels, so dass es keinen Konflikt gibt.

```
<key>MountEFI</key>
<string>Ja</string>
```

Dieser Parameter teilt dem Startskript mit, dass es ESP (EFI System Partition) Partition. Diese Option ist für die meisten Menschen unnötig oder temporär, sollten Sie Nein in die Konfig schreiben und ggf. Ja in das Menü eintragen. Ein weiterer möglicher Wert ist disk1, wenn Sie mehrere Festplatten haben und jede Festplatte ihre eigene Partition hat

```
EFI.
<key>LogEveryBoot</key>
```

<string>Ja</string>.

Das Download-Protokoll ist für Entwickler erforderlich, und normale Benutzer können setzen und Nein. Hier anstelle von Ja kann eine Zahl sein, wie viele Protokolle im System gehalten werden sollen.

<key>LogLineCount</key>
<string>3000</string>

die Anzahl der Zeilen in diesem Protokoll, dann werden die alten Zeilen durch die neuen ersetzt, so dass es keine unbegrenztes Wachstum dieser Datei.

Es ist nicht wirklich interessant, diese Protokolle zu akkumulieren. Sie können das letzte Protokoll immer mit dem Befehl:

\$ bdmesg > ~/Desktop/boot-log.txt erhalten.

DisableDrivers

```
<key>DisableDrivers</key>  
<array>  
  <string>CsmVideoDxe</string>  
  <string>VBoxExt4</string>  
</array>
```

Der Sinn dieses Abschnitts ist es, verschiedene config.plist in verschiedenen OEM-Ordnern zu haben, aber da der Treiberordner wird gemeinsam genutzt, müssen sie irgendwie unterscheiden, welcher Satz von Treibern auf einer bestimmten eine andere Konfiguration nutzt. Die eine braucht z. B. OsxAptioFixDxe, die andere braucht EmuVariableDxe. **Veraltet! Ein gemeinsamer Satz wird nun auf allen Boards verwendet.**

Quirks (Macken)

Die Geschichte dieses Abschnitts ist wie folgt. Beginn der Entwicklung von Clover für UEFI Boot war es Dmazar, der einen Treiber entwickeln sollte, um den Speicher zu korrigieren, der vom UEFI BIOS Aptio (American Megatrend) nicht mehr unterstützt wird. Der Punkt ist, dass die Allocate-Funktion in diesem BIOS weist unten Speicher zu, und um macOS zu starten, aber sie müssen den unteren Speicher frei halten. Der Konflikt betrifft nicht nur den Speicher, sondern auch die boot.efi Adressvirtualisierung, und es wirkt sich auf Zeiger, Funktionen usw. aus. (Nicht im Detail Ich schon?), Es war nicht meine Aufgabe, es war Dmazar, der Schritt für Schritt alle Konflikte gefunden hat und herausgefunden hat, wie man Erlaubnis einholt. Daraus wurde der Treiber OsxAptioFixDrv.efi. 32- und 64-Bit-Varianten mit allen Unterschieden

Adressierung.

Ich sollte anmerken, dass dieses Problem nicht mit dem Legacy Clover auftrat, denn in diesem Fall verwendet er Allocate von EDK2, und es belegt den Speicher von oben nach unten. Legacy Clover arbeitet ohne diese Driver.

Lange Zeit, nachdem Dmazar gegangen war, hat niemand diesen Treiber angerührt, außer vielleicht seine eigenen, separaten einzeiligen Ergänzungen wie Free2000. Und so übernahm vit9696 das Redesign dieses Treibers. Als erstes nahm er eine Änderung vor, um die Verwendung von natives NVRAM auf vielen Chipsätzen (BIOS) zu ermöglichen, mit denen es vorher nicht funktioniert hat.

A teilte er den Treiber weiter in sinnvolle Teile (Quirks) auf, die nun beliebig aktivieren und deaktivieren werden können, wenn der OpenCore-Lader verwendet wird. Aber auch ein Reddit-Dream-Programmierer hat sich auf den Weg gemacht und beschlossen, all diese Entwicklungen von OpenCore in einen separaten Treiber OcQuirks.efi, der zusammen mit dem Treiber OpenRuntime.efi funktioniert zusammen zu fassen, und alle Einstellungen werden in die Datei OcQuirks.plist geschrieben, um alle mit dem Clover-Bootloader zu nutzen.

Und dann ist da noch mein Umzug. Ich muss alle Quellen in einem Repo haben, um eine Halbierung des Quellcode durchzuführen. Und da die Lizenz für das Ganze Ihnen erlaubt, den Quellcode zu verwenden nach meinem eigenen Kopf, und historisch ist alles in seine historische Heimat zurückgekehrt, ich habe sie kopiert und aktualisiert, so dass anstelle einer separaten .plist-Datei wurde die gleiche clover config.plist verwendet, und diese Einstellungen können auch geändert werden nur von der GUI von Clover aus.



Das heißt, wenn Sie nicht sofort booten können, können Sie versuchen, zu diesem Menü zu gehen und um eine Art von Ja-Nein-Einstellung zu ändern. Nun die Details zu jedem Element. Es werden die Standardwerte angegeben.

```
<key>AvoidRuntimeDefrag</key>
<true/>
```

Verhindert die Speicherdefragmentierung für Runtime-Dienste wie die NVRAM-Unterstützung. Empfohlen für alle außer Apple und VMware.

```
<key>DevirtualiseMmio</key>
<false/>
```

Entfernt das Laufzeitattribut aus einigen bekannten MMIO-Bereichen. Nicht empfohlen für Systeme, die älter als Sandy Bridge sind. Die Liste der bekannten Regionen kann noch ergänzt werden im Abschnitt MmioWhitelist wählt aus. Nach meiner Beobachtung ist sie jedoch immer deaktiviert. Beansprucht eine Notwendigkeit für den Z390-Chipsatz.

```
<key>MmioWhitelist</key>
<array/>
```

Die Liste der Regionen ist als ein Array von Dictionaries definiert

```
<dict>
  <key>Comment</key>
  It's such a region.
  <key>Address</key>
  <string>0xffe00000</string>
  <key>Enabled</key>
  <true/>
</dict>
```

```
<key>DisableSingleUser</key>
<false/>
```

Verbietet die Verwendung des Befehlszeilenmodus, da dieser nicht sicher ist. ;)

<key>DisableVariableWrite</key>

<false/>

Verbietet den Schreibzugriff von McOS auf NVRAM, zur Sicherheit.

<key>DiscardHibernateMap</key>

<false/>

Verwenden Sie die alte Speicherkarte beim Aufwachen aus dem Ruhezustand. Verwendung, nur wenn Sie sicher sind, dass es bei ihnen der Fall ist. Es ist fast immer nicht der Fall.

<key>EnableSafeModeSlide</key>

<true/>

Standardmäßig ergibt das Booten im abgesicherten Modus slide=0. Dieser Patch ermöglicht einen anderen Wert, der in der Auswahl ProvideCustomSlide angegeben ist.

<key>ProvideCustomSlide</key>

<false/>

Ob die automatische Berechnung des slide=***-Wertes verwendet werden soll oder nicht. Der Bedarf ist aus dem Protokoll ersichtlich, wenn es eine OCABC-Meldung gibt: Nur N/256 Slide-Werte sind verwendbar!

<key>ProvideMaxSlide</key>

<integer>0</integer>

Wert von 1 bis 254, für den obigen Fall. Der Wert wird auf 255 gesetzt durch den Standardwerte, und dies kann zu Fehlern bei der Speicherzuweisung führen.

<key>EnableWriteUnprotector</key>

<true/>

Entfernt das Schreibschutz-Bit auf der Runtime services page . Aufgrund der Unsicherheit dieses Pick gibt es eine weitere RebuildAppleMemoryMap. Das heißt, für Systeme, die älter als 2018 sind, setzen wir:

EnableWriteUnprotector - True

RebuildAppleMemoryMap - False

SyncRuntimePermissions - False

Und bei neueren Geräten, die MATS unterstützen, wenn es eine solche ACPI-Tabelle im BIOS gibt

EnableWriteUnprotector - False

RebuildAppleMemoryMap - True

SyncRuntimePermissions - True

<key>ForceExitBootServices</key>

<false/>

Booten sie mit ExitBootServices bei neuen Speicherkarte. Dies ist der Punkt, an dem Kernel- und Kext-Patches auftreten. Verwenden sie es nur, wenn Sie sicher sind, was Sie tun.

<key>ProtectMemoryRegions</key>

<false/>

Ändert Attribute einiger Speicher-Regionen, wenn es einen Konflikt zwischen den Konzepten von macOS und dem BIOS gibt. Dieser Pick enthält mehrere Korrekturen, darunter eine von mir persönlich entwickelte, eine, die den CSM-Bereich schützt. Insbesondere, wenn Ihre Grafikkarte nicht über UEFI VBIOS verfügt, sollte dieser Pick aktiviert werden. (Im OsxAptioFix3Drv-Treiber ist es dagegen ist definitiv eingeschaltet und hat noch nie Beschwerden verursacht. Solange sie jedoch durch Standard.?)

<key>ProtectSecureBoot</key>

<false/>

Schützt die Verschlüsselungs-Schlüssel vor dem Überschreiben durch das Betriebssystem. Sichere Technik Boot, das wir irgendwie nicht besonders brauchen. Die Notwendigkeit dieses Picks wird beobachtet an einige Insyde BIOSe. Die anderen brauchen es nicht.

`<key>ProtectUefiServices</key>`

`<false/>`

Einige BIOSe wie VMware versuchen, Zeiger auf Runtime-Dienste umzuschreiben. Wir brauchen sie nicht, also schützen wir sie. Jemand namens Rediskin behauptet, es sei für Z390-Chipsatz notwendig.

`<key>ProvideConsoleGopEnable</key>`

`<true/>`

Erzeugt ein GOP-Protokoll für den Konsolenmodus, d. h. für Textausgaben, die nicht in Text Modus, wie Sie es aus dem PC-BIOS gewohnt sind, sondern im grafischen Modus, wie es Apple macht. (diese Auswahl ist unbedingt erforderlich)

`<key>RebuildAppleMemoryMap</key>`

`<false/>`

Erzeugen sie eine macOS-kompatible Speicherkarte. Die Sache ist die, dass Apple eine etwas andere Ideen hat, wie und was zu tun ist, in unserem UEFI-BIOS. Das Problem ist jedoch, dass unsere die Speicherkarte zu unserer Hardware passt. Deshalb schalten wir die Auswahl ab. Die anderen Picks machen einen notwendigen Teil der Arbeit. Dieser Pick ist eine Art Ersatz EnableWriteUnprotector für Systeme, die MAT unterstützen. Für diese Auswahl muss auch SyncRuntimePermissions aktiviert sein. Wir haben es jedoch in dem Standard. Siehe EnableWriteUnprotector pick

`<key>SetupVirtualMap</key>`

`<true/>`

Diese Einstellung befasst sich mit der Reihenfolge, in der virtuelle Adressen zugewiesen und verwendet werden. Einige BIOSe, wie z. B. OVMF, unterstützen diesen Pick nicht. Hmm, warum sollte das OcQuirks.efi-Treiber auf einem System mit OVMF?! Er funktioniert auch ohne ihn, genau wie der alte Clover. Schalten wir es also ein.

`<key>SignalAppleOS</key>`

`<false/>`

Sagt dem BIOS, dass wir ein macOS-System laden, obwohl wir Windows laden. Es muss bei einigen Macbooks gemacht werden, aber nicht wir. Wird vielleicht von OpenCore benötigt, aber nicht von Clover.

`<key>SyncRuntimePermissions</key>`

`<true/>`

Aktualisiert die Berechtigungsflags im Bereich Runtime. Natürlich brauchen sie das.

Die allgemeine Situation ist, dass für meine nicht mehr ganz so neuen Computer diese speziellen Einstellungen die Vorgaben ausreichend waren. Dass Rediskin eine andere Liste hat, und er behauptet, dass diese Liste entspricht dem Verhalten von AptioMemoryFix. Dies ist jedoch nicht der Fall mMit seinem Set. Mein Computer bootet nicht, während mit dem alten AptioMemoryFix alles in Ordnung ist. Also der Standardsatz von Clover unterscheidet sich von den ursprünglichen OcQuirks (RIP). In Clover 5125 sind durch die Aufnahme von Patches aus OpenCore in diesem Abschnitt neue Werte vorhanden.

`<key>FuzzyMatch</key>`

`<false/>`

Der Schlüssel gewährleistet die Kompatibilität mit System 10.6 (Snow Leopard), da die Menge des Cache unterschiedlich berechnet wird. Ich weiß nicht, wie es mit OpenCore aussieht, aber Clover wird schon seit langem Schneeleo geladen, zusammen mit dem gleichen Cache.

```
<key>KernelCache</key>
<string>Auto</string>
```

Mögliche Werte (Auto, Cacheless, Mkext, Prelinked). Diese Cache-Typen waren vor dem System 10.7 aktuell und wurden üblicherweise mit dem Boot-Argument Kernel=, Mkext= und anderen ausgewählt, die es im alten Clover gab.

```
<key>AppleXcpmExtraMsrs</key>
<false/>
```

XCPM-Unterstützung auf redundanten CPUs, etwas, was der KernelXCPM-Patch früher getan hat.

```
<key>AppleXcpmForceBoost</key>
<false/>
```

Dieser Patch schreibt 0xFF00 in das Register MSR_IA32_PERF_CONTROL = 0x199, d. h. statt normaler p-Zustände bietet OpenCore an, auf den maximalen Schritt zu verbieten.

```
<key>DisableIoMapper</key>
<true/>
```

Deaktiviert die VT-d-Technologie auf eigene Art und Weise. Es gibt eine Alternative, um die DMAR-Tabelle zu verwerfen. Und dann gibt es noch die Option dart=0.

```
<key>DisableLinkeditJettison</key>
<true/>
```

Erlaubt Lilu.kext, schneller zu laufen. Auf Kosten dessen, was unbekannt ist. Und wie keepsyms=1 mehr nicht erforderlich.

```
<key>DisableRtcChecksum</key>
<false/>
```

Der Sinn des Patches ist, dass die RTC im Mac anders verwendet wird als im PC, und die Prüfsumme wird schlecht. Wir blockieren einfach seine Zählung. Es gibt ein AppleRTC=YES-Analogon in Clover, und wir tun es.

```
<key>DummyPowerManagement</key>
<false/>
```

Blockiert AppleIntelCpuPowerManagement als Alternative zum NullCPUPM Kext. Zur Erinnerung, dieses Kext wird meistens blockiert, wenn 0xE2 gesperrt ist, und es gibt andere Patches dagegen.

```
<key>ExternalDiskIcons</key>
<false/>
```

Der altbekannte Aufnäher gegen gelbe Icons ist als Spitzhacke ausgeführt.

```
<key>IncreasePciBarSize</key>
<false/>
```

Erhöht den Speicherwert für die Leiste von 1 GB auf 4 GB im Gütesiegel IOPCIFamily. Ich weiß nicht, wer sich diesen Patch ausgedacht hat und für welche Fälle. OpenCore rät davon ab, ihn überhaupt zu verwenden.

```
<key>PowerTimeoutKernelPanic</key>
<false/>
```

Dieser Patch, der nur für Catalina ist, verhindert, dass der Kernel in Panik gerät, wenn ein Gerät sehr lange braucht um zu reagieren. Warum stornieren Sie es?! Lassen Sie es in Panik geraten, anstatt stumm zu schweben. Wenn Sie jemals jemals auf eine solche Situation stoßen, versuchen Sie diesen Patch.

`<key>ThirdPartyDrives</key>`

`<false/>`

Ein seit langem bekannter Patch ist in der Clover-Konfiguration zu finden: Enable Trim on Non-Apple. Es geht darum, das Wort Apple in einer Leerzeile zu ersetzen. Es gibt auch einen Kommentar, dass Catalina intern eine Gelegenheit bietet, das Gleiche zu tun.

`<key>XhciPortLimit</key>`

`<true/>`

Es ist ein bekanntes Problem, dass Apple die Anzahl der XHCI-, USB2+USB3-Controller-Ports begrenzt auf die Anzahl von 15, und obwohl die neuen Chipsätze mehr Anschlüsse bieten, hat es Apple nicht eilig, diesen Grenzwert erhöhen. Binäre Patches gibt es schon lange, aber sie sind abhängig von der Version der Systeme. Inwieweit es OpenCore gelungen ist, die Versionsabhängigkeit zu überwinden, weiß ich nicht. Die Praxis zeigt an, dass alle nicht entfernbaren Patches entfernt werden. Ich habe mir eine Legacy_USB.kext erstellt, der die USB2-Ports entfernt, die nicht benötigt wurden, und dadurch Platz für USB3 machte. Dieser Kext funktionierte noch in 10.13, und auch noch in 11.1. arbeitet

Acpi

Eine Gruppe von Parametern, die die Korrektur von verschiedenen ACPI-Tabellen regeln. Und es ist nicht nur so, dass der Mac seine eigenen Anforderungen hat, es sind einfach unterschiedliche Versionen des ACPI Spezifikationen und elementare Faulheit der Hersteller, und es gibt einfach im BIOS des Motherboards keine Informationen über die installierten Karten und die CPU (können Sie diese nicht dynamisch erkennen? Clover (Kleeblätter) tut es!).

Parameter für die FADT-Tabelle

`<key>ResetAddress</key>`

`<string>0x64</string>`

`<key>ResetValue</key>`

`<string>0xFE</string>`

Diese beiden Parameter dienen für einen sehr wertvollen Fix, dem Neustartfix. Diese Werte sollten in der FADT-Tabelle stehen, aber aus irgendeinem Grund sind sie nicht immer da, Die Tabelle selbst ist vielleicht kürzer als nötig, so viel kürzer, dass diese Werte verworfen werden. Standardmäßig geht der bereits in der FACP vorhandene Wert, wenn jedoch nichts, dann wird das Paar 0x64/0xFE verwendet, was einen Neustart über den PS2-Controller bedeutet. Die Praxis hat gezeigt, dass dies nicht bei jedem funktioniert, ein anderes mögliches Wertepaar ist 0x0CF9/0x06, was einen Neustart über den PCI-Bus bedeutet. Dieses Paar wird auch auf dem Original Mac verwendet, aber nicht immer funktioniert es auf Hackintoshes. Der Unterschied ist klar, es gibt auch einen PS2-Controller auf Hackintoshes, der einen Neustart verhindern kann, wenn er nicht zurückgesetzt wird. Eine andere Möglichkeit ist 0x92/0x01, ich weiß es nicht, vielleicht hilft es ja jemandem.

`<key>HaltEnabler</key>`

`<true/>`

Und dies ist ein Fix, der das Shutdown/Sleep-Problem beim UEFI-Boot löst. Der Fix wird einmalig vor dem Aufruf von boot.efi durchgeführt, ist also nicht 100%ig effektiv garantiert. Trotzdem ist es, zumindest auf Intel-Chipsätzen, recht sicher. Für mich persönlich ist es ein Lebensretter!

`<key>smartUPS</key>`

`<false/>`

Eigentlich ist dieser Parameter dazu gedacht, ein Profil mit der Leistung=3 in der FADT-Tabelle vorzuschreiben. Die Logik ist wie folgt:

PM=1 - Desktop , Netzversorgung

PM=2 - Notebook, netz- oder batteriebetrieben

PM=3 - Server, versorgt durch SmartUPS, von das auch MacOSX kennt.

Die Wahl zwischen 1 und 2 trifft Clover auf der Grundlage einer Analyse des Mobilitätsbits, aber es gibt auch Mobile-Parameter im Abschnitt SMBIOS. Sie könnten z. B. sagen, dass wir einen MacMini haben, und dass er mobil sein soll. Der Wert 3 wird ersetzt, wenn smartUPS=Yes.

MADT-Abgleich (APIC)

```
<key>PatchAPIC</key>
```

```
<false/>
```

Auf manchen Rechnern ist es nur möglich, das System mit cpus=1 zu booten, oder mit einem speziellen gepatchten Kernel (Lapic NMI Patch). Eine einfache Analyse zeigte, dass sie die falschen MADT-Tabelle benutzen, d. h., es sind keine NMI-Partitionen darin enthalten. Dieser Parameter dient dazu solche Tabellen im laufenden Betrieb anzupassen. Gegen einen gesunden Computer ist nichts einzuwenden passieren wird. Allerdings habe ich auch keine Berichte gesehen, die irgendjemandem geholfen haben. Ja, es gibt Entwickler, der es braucht, und der diesen Patch in neuen Versionen von Clover korrigiert hat. Es gibt einen entsprechenden Patch im Abschnitt KernelAndKextPatches, der auch dieses Problem löst, aber mit anderen Mitteln.

Andere ADC-Tabellen:

```
<key>DropTables</key>
```

```
<array>
```

```
  <dict>
```

```
    <key>Signature</key>
```

```
    <string>DMAR</string>
```

```
  </dict>
```

```
  <dict>
```

```
    <key>Signature</key>
```

```
    <string>MCFG</string>
```

```
  </dict>
```

```
    <key>Signature</key>
```

```
    <string>SSDT</string>
```

```
    <key>TableId</key>
```

```
    <string>CpuPm</string>
```

```
    <key>Length</key>
```

```
    <string>0x0fe1</string>
```

```
  </dict>
```

```
</array>
```

In diesem Array listen wir die Tabellen auf, die wir verwerfen wollen.

DMAR - weil Mac nicht mit der VT-d-Technologie kompatibel ist. Oder besser gesagt, es hat eine andere Tabelle. **MCFG** - denn durch die Angabe eines MacBookPro oder MacMini erhalten wir eine brutale Bremsen. Es wurde aber bereits eine bessere Methode entwickelt.

```
<key>FixMCFG</key>
```

```
<true/>
```

In diesem Fall wird die Tabelle nicht verworfen, sondern korrigiert. Der Autor des Patches ist wieder vit9696. Dennoch bleibt die Methode zum Verwerfen dieser Tabelle bis auf weiteres auf Lager.

Zurück zur Geschichte von DropTables

SSDTs sind unterschiedlich, und wir geben zusätzlich TableId an, welche wir verwerfen werden, weil wir unsere SSDT-Tabellen nach den Regeln von Apple erstellen werden, und

Nicht Gigabyte, oder, Gott bewahre, ASUS. Sie können in der Kopfzeile der Tabelle nachsehen, oder im Clovers Boot Logbuch. Hier ist zum Beispiel eine Tabelle, die nicht verworfen werden sollte. DefinitionBlock ("SSDT-0.aml", "SSDT", 1, "SataRe", "SataTabl", 0x00001000) Für die gespeicherten Tabellen gilt die Regel für binäre gepatchte DSDT, d.h. diese Tabellen werden ebenfalls geändert, was logisch ist. Wenn alle SSDT-Tabellen aus irgendeinem Grund die gleiche TableID haben, können Sie die Länge der Tabelle, die gelöscht werden soll angeben. Die Länge kann wie oben in Hex eingestellt werden, oder sie kann in <ganzzahlig> als Dezimalzahl.

```
<key>DisableASPM</key>
```

```
<false/>
```

Dies betrifft die ACPI-Einstellungen des Systems selbst, so dass Apples ASPM-Management funktioniert nicht wie vorgesehen. Zum Beispiel ein nicht-nativer Chipsatz. In welchen Fällen anzuwenden, und was es bewirkt, weiß ich nicht mehr.

```
<key>SSDT</key>
```

```
<key>DropOem</key>
```

```
<true/>
```

Da wir unsere SSDT-Tabellen erstellen oder dynamisch laden werden, müssen Sie unnötige Überschneidungen von Interessen vermeiden. Mit diesem Parameter verwerfen Sie alle nativen Tabellen, zugunsten von neuen Tabellen. Oder wollen Sie kategorisch vermeiden, die Tabellen zu patchen SSDT. Sie haben folgende Möglichkeit: Legen Sie die nativen Tabellen mit kleinen Bearbeitungen in den Ordner EFI/OEM/xxx/ACPI/patched/, und lassen Sie die ungepatchten weg. (igitt!) ungepatchte Tabellen verwerfen. Es ist jedoch besser, die oben erwähnte Methode des selektiven Verwerfens zu verwenden.

```
<key>Generate</key>
```

```
<dict>
```

```
  <key>CStates</key>
```

```
  <true/>
```

```
  <key>PStates</key>
```

```
  <true/>
```

```
</dict>
```

Hier definieren wir, dass zwei zusätzliche Tabellen für C-Zustände und für P- erzeugt werden nach den von der Hack-Community entwickelten Regeln. Für C-Zustände, Tabelle mit den Parametern C2, C4, C6, Latenz, die im Abschnitt CPU genannt werden. Es ist auch möglich, Parameter im SSDT-Abschnitt anzugeben, was logisch ist

```
  <key>EnableC7</key>
```

```
  <true/>.
```

```
  <key>EnableC6</key>
```

```
  <true/>.
```

```
  <key>EnableC4</key>
```

```
  <false/>.
```

```
  <key>EnableC2 </key>
```

```
  <false/>
```

```
  <key>C3Latency</key>
```

```
  <integer>67</integer>
```

Die Tatsache, dass diese Generierung wirksam geworden ist, wird durch das Kernel-Protokoll kontrolliert. Ohne diese Methode ACPI_SMC_PlatformPlugin::pushCPU_CSTData - _CST-Auswertung würden fehlgeschlagen. Ein gesondertes Wort zur C3Latency. Dieser Wert erscheint in echten Macs, für iMacs in der Größenordnung von 200, für den MacPro in der Größenordnung von 10. Meiner Meinung nach werden iMacs durch P-Zustände geregelt, MacPro - D-Zustände. Und es hängt auch vom Chipsatz ab, ob Ihr Chipsatz angemessen auf

C-State-Befehle von MacOS reagiert. Die einfachste Möglichkeit ist, diesen Parameter nicht zu schreiben, und alles wird so funktionieren, wie es ist. Für P-Zustände gibt es eine Tabelle, die den Prozessorteil um die Methoden `_PPC`, `_PCT` und `_PSS` ergänzt.

`_PCT` - Performance Control - Steuerung des speedstep control (der Fahrstufenregelung).

`_PPC` - Performance Present Capabilities, Diese Funktion liefert eine Zahl, die eine Frequenzgrenze bedeutet. Details unten, im Parameter `PLimitDict`.

`_PSS` - Performance Supported States - Ein Set der möglichen Prozessorzustände – P-States. Dieses Array wird aus den Prozessordaten erzeugt, die Clover bereits berechnet hat, und auch unter Berücksichtigung der Parameter des Anwenders:

`<key>PLimitDict</key>`

`<string>1</string>`

Die Essenz des Parameters ist sehr einfach - die maximale CPU-Frequenz zu begrenzen. Ein Wert von 0 ist. Betrieb bis zum Maximum, 1 - ein Schritt weniger als das Maximum, 2 - zwei Schritte. Beispiel:

Core2Duo T8300 2400MHz läuft mit einer maximalen Frequenz von 2000, wenn er auf zwei Schritte eingestellt wird. Warum sollten sie das machen? Um eine Überhitzung des Laptops zu vermeiden, sind die Fähigkeiten der CPU die Kühlleistung überschreiten begrenzt. Der exakt gleiche Parameter ist in z.B. Plattformplatten: `System/Library/Extensions/IOPlatformPluginFamily.kext/Contents/PlugIns/ACPI_SMC_PlatformPlugin.kext/Contents/Ressourcen/MacBook5_1.plist`

Wir werden diese Plists weiter unten besprechen.

Bei einigen Prozessoren, z. B. Core2Quad, wird festgestellt, dass `PLimitDict` entgegengesetzt funktioniert , und bester Fall =1 ist. Es ist durchaus möglich, dass es sich nur um einen Fehler in der DSDT handelt. Zum Beispiel, weil sie den Darwin-Patch nicht machen wollten

`<key>UnderVoltStep</key>`

`<string>1</string>`

Optionaler Parameter zur Reduzierung der Prozessortemperatur durch Absenken der Betriebsspannung. Mögliche Werte sind 0, 1, 2, 3 ... je mehr, desto mehr kühlen wir die CPU ab, bis der Computer sich aufhängt. Hier funktioniert die Überlastungssicherung, bei Clover wird nicht es Ihnen erlaubt, den Wert außerhalb des erlaubten Bereichs zu setzen, bzw. zu schreiben, was Sie wollen, und Nur was erlaubt ist, funktioniert. Die zulässigen Werte können aber auch zu instabiler Betrieb. Die Wirkung dieses Parameters wird tatsächlich beobachtet. Allerdings nur für Perinna.

`<key>DoubleFirstState</key>`

`<true/>`

Ich habe festgestellt, dass es für einen erfolgreichen Speedstep notwendig ist, den ersten in der P-States-Tabelle zu duplizieren Zustand. Nach der Einführung anderer Parameter ist die Notwendigkeit dafür fraglich geworden. Es ist auch ist nur für IvyBridge wahr, der Rest hebt sich bedingungslos auf.

`<key>MinMultiplier</key>`

`<integer>7</integer>`

Minimaler CPU-Multiplikator. Er selbst meldet 16 und bevorzugt den Betrieb mit Frequenz von 1600, für Speedstep sollten Sie jedoch die Tabellenwerte auf 800 oder sogar 700 stellen. Empirie.

`<key>MaxMultiplier</key>`

`<integer>30</integer>`

Eingeführt in Analogie zum Minimum, aber es scheint vergeblich zu sein. Sie sollte nicht eingefügt werden. Das ist jedoch der Fall. irgendwie die Anzahl der P-Zustände beeinflusst, so dass Sie experimentieren können, jedoch ohne, daß es dafür keine besondere Notwendigkeit gibt, das zu tun.

```
<key>Generate</key>
  </dict>
    <key>C-Staaten</key>
    <true/>
    <key>PluginTyp</key>
    <false/>
    <key>APLF</key>
    <false/>
    <key>APSN</key>
    <false/>
    <key>PStaaten</key>
    <true/>
  </dict>
```

Im neuen Clover ist diese Gruppe von Parametern in einem Abschnitt zusammengefasst, und PluginType ist nun nur wahr oder falsch, weil es keine anderen Möglichkeiten gibt. APLF- und APSN-Parameter beeinflussen den Speedstep, aber für diejenigen, die wissen, was sie machen.

```
<key>PluginType</key>
<integer>0</integer>
```

Für IvyBridge-, Haswell- (und höhere?) Prozessoren sollten Sie 1 eintragen, für den Rest 0. Dieser Schlüssel erzeugt zusammen mit dem Schlüssel Generate – PluginType die SSDT-Tabelle, enthält nur PluginType, aber keine P-States, wenn deren Erzeugung deaktiviert ist.

Großer Abschnitt über DSDT-Setup und Patches.

```
<key>DSDT</key>
<dict>
  <key>Debug</key>
  <false/>
```

Mit diesem Parameter können wir sehen, was mit dem DSDT passiert, während es gepatcht wird, wenn dadurch das System am Hochfahren gehindert wird. Zuerst wird die Originalversion gespeichert /EFI/CLOVER/ACPI/origin/DSDT-or.aml, dann gehen Sie die Prozedur aller Patches durch (Übrigens es hinterlässt eine ganze Reihe von Meldungen in debug.log, wenn es auch verbunden ist), und dann speichert es die Datei /EFI/CLOVER/ACPI/origin/DSDT-pa0.aml, wenn eine solche Datei bereits existiert mit des vorherigen Versuchs wird die nächstnumerierte DSDT-pa1.aml erstellt, DSDT- pa2.aml... , überschreiben sie sich nicht gegenseitig. Denken Sie am Ende aller Übungen daran den Ordner zu leeren.

```
<key>Name</key>
<string>DSDT.aml</string>
```

Sie können mehrere Versionen der DSDT-Datei mit beliebigem Namen haben, und Sie können einen nicht existierenden Namen schreiben, z. B. Standard-BIOS.aml, und dieser wird dann als Basis genommen für die DSDT, die im BIOS verfügbar ist. Die Dateiprioritäten sind wie folgt:

1. Die höchste Priorität hat die Datei DSDT.aml, die im Stammverzeichnis des gebooteten Systems liegt. Die Logik ist, um verschiedene Computer mit demselben Flash-Laufwerk zu booten, von denen jeder seine eigene DSDT braucht.

2. Wenn es keine solche Datei gibt, suchen Sie auf dem Flash-Laufwerk im OEM-Bereich: /EFI/CLOVER/OEM/p8b/ACPI/patched/DSDT.aml

3. wenn es auch dort nicht zu finden ist, schauen Sie in den gemeinsamen Ordner /EFI/CLOVER/ACPI/patched/DSDT.aml

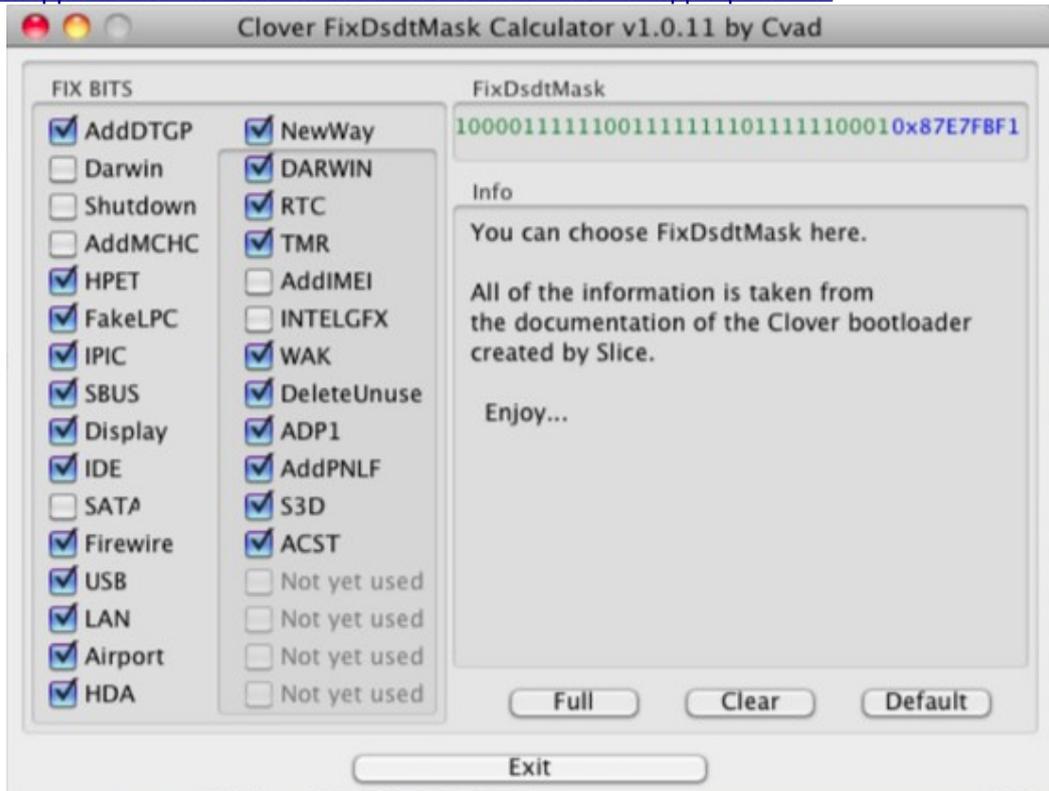
<key>FixMask</key>

<string>0xFFFFFFFF</string>

Dieser Parameter blendet 32 Patches auf einmal für die DSDT-Tabelle aus, und zwar um die Anzahl der Bits in die Maske.

Um zu berechnen, wie sich die Summe der Bits zu einer bestimmten Maske addiert, können sie den Systemrechner nehmen, stellen sie ihn in die Programmiereransicht und schalten Sie auf 16 Zahlen. Und nun geben wir durch Anklicken der Bits 0 bis 31 die gewünschte Maske ein. Es gibt eine offensichtlichere Variante: CloverFixDsdtdMaskCalculator von Cvad

<http://www.applelife.ru/attachments/cloverfixdsdtmaskcalculator-app-zip.43973/>



Ab der Revision 2184 können (und sollten) Patches nach und nach in den folgenden Abschnitt eingefügt werden

<key>Fixes</key>

<dict>

<key>AddDTGP</key>

<true/>

<key>FixDarwin</key>

<true/>

<key>FixShutdown</key>

<true/>

<key>AddMCHC</key>

<false/>

<key>FixHPET</key>

<true/>

<key>FakeLPC</key>

<false/>

<key>FixIPIC</key>

<true/>

<key>FixSBUS</key>

<true/>

```

<key>FixDisplay</key>
<true/>
<key>FixIDE</key>
<false/>
<key>FixSATA</key>
<false/>
<key>FixFirewire</key>
<true/>
<key>FixUSB</key>
<true/>
<key>FixLAN</key>
<true/>
<key>FixAirport</key>
<true/>
<key>FixHDA</key>
<true/>
<key>FixDarwin7</key>
<true/>
<key>FixRTC</key>
<true/>
<key>FixTMR</key>
<true/>
<key>AddIMEI</key>
<true/>
<key>FixIntelGFX</key>
<false/>
<key>FixWAK</key>
<true/>
<key>DeleteUnused</key>
<true/>
<key>FixADP1</key>
<true/>
<key>AddPNLF</key>
<true/>
<key>FixS3D</key>
<true/>
<key>FixACST</key>
<true/>
<key>AddHDMI</key>
<true/>
<key>FixRegionen</key>
<true/>

```

</dict>

Wenn dieser Fixabschnitt vorhanden ist, wird der maskierte Fixschlüssel ignoriert. Aber um Ihnen zu sagen, was diese Korrekturen bedeuten, müssen Sie ein neues Kapitel aufschlagen. Weitere Keys, um einige Probleme mit dem automatischen Patching zu lösen.

<key>ReuseFFFF</key>

<false/>

In einigen Fällen wird der Versuch, die Anzeige zu patchen, durch das Vorhandensein der ursprünglichen DSDT-Geräte des Typs
Gerät (PEGP)

```

{
  Name (_ADR, 0xFFFFFFFF)
  Name (_SUN, Eins)
}

```

Sie können seine Adresse auf 0 ändern, aber das funktioniert nicht immer. <true/> - versuchen zu ändern Adresse, <false/> - gehen Sie weg und versuchen Sie nicht, es zu patchen.

Dieser Fix ist im neuen Clover ab 5116 ausgeschlossen.

<key>DropOEM_DSM</key>

```

    <Diktat>.
    <key>ATI</key>.
    <true/>.
    <key>NVidia</key>
    <true/>.
    <key>IntelGFX</key>
    <true/>.
    <key>HDA</key>
    <true/>.
    <key>HDMI</key>
    <true/>.
    <key>LAN</key>
    <true/>.
    <key>WIFI</key>
    <true/>.
    <key>SATA</key>
    <true/>.
    <key>IDE</key>.
    <true/>.
    <key>USB</key>
    <true/>.
    <key>LPC</key>
    <false/>.
    <key>SmBUS</key>
    <false/>.
    <key>Feuerdraht</key>
    <true/>.
</dict>.

```

In einigen Fällen hat das Gerät, das wir automatisch patchen wollen, bereits OEM-Methode `_DSM`. Sie können nicht duplizieren, daher gibt es zwei Möglichkeiten:

```

<true/> - die ursprüngliche Methode wird verworfen und stattdessen wird unsere Methode erzeugt,
<false/> - beim Auftreten der ursprünglichen Methode ziehen wir uns zurück, ohne etwas zu tun. Was ist an
der ursprünglichen Methode dran? Kaum das, was wir gerne sehen würden, und kaum das, was wird für
OSX benötigt. Normalerweise denken BIOS-Hersteller nur an Windows. Aber wenn Sie denken, dass es
etwas Wichtiges in dieser Methode gibt, dann injizieren Sie Ihre Eigenschaften in dieses Gerät mit Strings
(siehe Kapitel Geräte->Injizieren). Ich würde also empfohlen, alle diese OEM-DSMs zu verwerfen, außer
wenn Sie setzen Sie Ihr benutzerdefiniertes DSDT, und wenden Sie weitere Korrekturen aus dem Bereich
automatisch, aber Sie sind nicht bereit, Ihre _DSM-Methoden für automatisch zu ersetzen generiert.

```

Das DSDT meines neuen Computers erwies sich als wahnsinnig schlecht in Bezug auf macOS. Es hat vielen DSM-Methoden, sowohl in der Elternschaft als auch bei den Töchtern. Das Ergebnis ist Panik. DropOEM_DSM ist hier machtlos. Der folgende Patch hat geholfen Patches → find: `_DSM`, ersetzen: `ZDSM`. Siehe unten.

```

<key>SuspendOverride</key>
<false/>

```

Der Shutdown-Patch funktioniert nur bei Zustandstyp 5 - Shutdown, wir möchten jedoch diesen Patch auf die Zustände 3 und 4 erweitern, dann setzen Sie `SuspendOverride = true`. Es half mir beim Sleep-Problem während eines UEFI-Boots. Andernfalls würde der Bildschirm erlöschen und die Beleuchtung und die Ventilatoren liefen weiter.

Fortgeschrittene Hacker können ihre eigenen DSDT-Patches erstellen, indem sie einen Ersatz für Binärebene:

```

<key>Patches</key>
<array>
    <dict>

```

```

    <Taste>Finden</key>
    <data>UFhTWAhfQURSAAhfUFJXEgYC</data>
    <key>Replace</key>
    <data>UFhTWAhfQURSAAhfU1VOCgQIX1BSVxIGAg==</data>
    <key>TgtBridge</key>
    <data>UFhTW</data>
  </dict>
</dict>
  <key>Comment</key>
  <string>Rename oem _DSM to ZDSM</string>
  <key>Disabled</key>
  <false/>
  <key>Find</key>
  <data>X0RTTQ== </data>
  <key>Replace</key>
  <data>WkRTTQ== </data>
</dict>
</array>

```

TgtBridge definiert, dass das Patch nur innerhalb eines einzelnen Blocks funktioniert, der durch unter diesem Namen Gerät (RP02) { . } definiert wird.

Diese Methode ist jedoch ungenau, da Blöcke fragmentiert sein können, es ist besser, sie verwenden die Methode RenameDevices (s. unten)

Die spezifischen Zahlen sind von Ihren Entwürfen, wenn Sie wissen, was zu tun ist. Wenn die Stringlängen nicht übereinstimmen, wird Clover die Längenänderung korrekt berücksichtigen, mit einer Ausnahme: Wenn es innerhalb einer If- oder Else-Anweisung geschieht. Wenn Sie eine solche Änderung wünschen, ersetzen Sie den gesamten Operator. (The length of the strings may not be the same, Clover will correctly consider the length change, with one exception: to prevent it from happening inside the If or Else statement.)

Einige Erklärungen. Die Kommentartaste dient nicht nur dazu, Sie daran zu erinnern, dass hier ein Kommentar geschrieben steht, sie wird auch im Clover-Menü verwendet, um Menüs durch connect/disable dieser Fixes zu compilieren. Den Anfangswert freigegeben oder nicht wird durch die Zeilen Disabled=false definiert. Der Standardwert ist aktiviert. Wenn Sie Patches schreiben, ist es besser, sie zuerst als Disabled=true zu schreiben, und dann im Menü ihre jeweils einen zuzulassen.

Andere ACPI tables

Clover verfügt über einen Dienst zum Laden anderer Tabellen. Insbesondere ist es sehr häufig eine ganze Bibliothek mit verschiedenen SSDT-xxx.aml für verschiedene Geräte und für Speedstep zu erstellen. Ab der Revision 3088 hat sich die Regel zum Laden solcher Tabellen geändert.

Legen Sie die Tabellen in den Ordner
/EFI/CLOVER/OEM/xxx/ACPI/patched/

Wenn es keinen solchen Ordner gibt, wird der öffentliche Ordner als /EFI/CLOVER/ACPI/patched/ verwendet. Alle Dateien mit der Erweiterung ".aml", die nicht mit einem Punkt "." beginnen, und nicht die Zeichenfolge "DSDT" in seinem Namen enthält, werden aus diesem Ordner geladen, weil die verschiedenen DSDTs die Optionen mit einem anderen Algorithmus geladen werden. Die Bootreihenfolge ist nicht garantiert. Wenn wir eine streng definierte Reihenfolge haben wollen beim booten, müssen wir es explizit in die Config schreiben:

```

<key>ACPI</key>
<dict>
  <key>SortedOrder</key>
  <array>
    <string>SSDT-3.aml</string>

```

```

    <string>SSDT-1.aml</string>
    <string>SSDT-2.aml</string>
  </array>

```

Und wenn ein solches Array vorhanden ist, werden nur diese Tabellen geladen, und zwar genau in dieser Reihenfolge (lacht) Okay?).

Ein weiteres Problem bei Tabellen, ist der Name. Der OEM wird weder durch die Verwendung der nationalen Alphabet, noch das Fehlen eines Namens verboten, aber für Apple ist dies inakzeptabel. Der Name muss 4 des lateinischen Alphabets sein. Der nächste Fix wird das beheben.

```

<key>FixHeaders</key>
<true/>

```

Genau derselbe Fix war jedoch im Abschnitt DSDT Fixes, da er sich nicht auf DSDT bezieht, wird er in einem separaten Punkt des ACPI-Abschnitts abgelegt.

```

<key>RenameDevices</key>
<dict>
  <key>_SB.PCI0.RP02.PSXS</key>
  <string>ARPT</string>
  <key>_SB.PCI0.EHC1</key>
  <string>EH01</string>
  <key>_SB.PCI0.POP2.PEGP</key>
  <string>GFX0</string>
</dict>

```

Anders als ACPI-Binärpatches, die übrigens nicht nur DSDT, sondern auch SSDT als auch, sowohl nativ als auch bootfähig, dient diese Methode als Ersatz für einen Patch wie Suchen->PX SX, Ersetzen->ARPT. Wenn aber im Abschnitt DSDT->Patches eine solche Ersetzung auf in der Methode RenameDevices beruht, sucht der Algorithmus nur nach diesen Geräten, die auf der besagten Bridge liegen.

Hier ist ein komplexes Beispiel zum Ersetzen von `_SB.PCI0.RP02.PSXS` → `ARPT Scope(_SB)`

```

{
  Device (PCI0)
  {
    Device (RP02)
    {
      Device (PSXS) <- hier ändern
      {
        Method (_ON)
        {
        }
      }
      Method (_OFF)
      {
      }
    }
    PSXS._ON() <- hier ändern
  }
  Scope (RP02)
  {
    PSXS._OFF() <- hier ändern
  }
  Device (RP03)
  {
    Device (PSXS) <- hier nicht ändern
    {
    }
  }
  PSXS._ON() <- hier nicht ändern
}

```

DSDT adjustment

DSDT - Differentiated System Description Table - die größte und komplexeste ACPI Tabelle. Die Mindestlänge beträgt 36 Byte, die tatsächliche Länge beträgt 20kb oder mehr. Diese Tabelle beschreibt Geräte und deren Zugriffsmethoden. Zugriffsmethoden können arithmetische und logische Ausdrücke sein, und bilden somit ein Programm in einer Programmier-Sprache ähnlich wie in C durch die geschweiften Klammern. Um diese Tabelle zu korrigieren bedeutet, dass Sie etwas über Programmierung wissen müssen. Clover bietet eine Art von Variante der automatischen Bearbeitung, aber Sie müssen verstehen, dass es in Clover noch keine künstliche Intelligenz gibt, und die automatischen Programmanpassungen sind noch lange nicht perfekt. Ein Programmierer*in wird es besser machen.

Warum muss es repariert werden? Das gesamte DSDT-Patch war von Anfang an darauf ausgerichtet zuallererst das HPET-Gerät - High Precision Events Timer - zu reparieren. Der Punkt ist, dass das OSX-System das Paket AppleIntelCPUPowerManagement enthält, das zur Steuerung der Prozessorleistung (Speedstep des Prozessors) dient, und die strikt verlangt, dass dieses Gerät war ein HPET mit IRQ-Interrupts. Ohne diese Bedingung gerät der Kext in Panik.

Es ist nur möglich, durch Deaktivieren oder Löschen dieses Kext zu arbeiten. Aber es gibt noch eine andere Möglichkeit. Korrigieren Sie die DSDT und der HPET startet richtig! Jedoch ab MacOS 10.9 haben sich Anforderungen geändert. Blödsinn, der Patch ist weiterhin erforderlich und hat die gleiche Gültigkeit. Im neuen Systemen, auf der neuen Hardware hat bereits er seine Relevanz verloren, vor allem aufgrund der Tatsache, dass die Kext-Funktionalität in den Kernel übergegangen ist.

Dies ist Patch **Nr. 1**, eine lebensnotwendige Notwendigkeit. Ist MacOS das einzige, das diesen HPET benötigt? Sicher nicht, aber die BIOS-Hersteller beginnen gerade erst, dies zu erkennen und schreiben korrekten Parametern, ist es immer noch selten, dass ein DSDT ohne Patch funktioniert.

Moment **Nr. 2**. Im DSDT lassen sich einige betriebliche Abhängigkeiten erkennen Systeme, "Windows 98", "Windows 2001", "Windows 2006 (Vista)", "Linux", MacOS hat „Darwin“-Kennung, und der Mit dem Mainboard ausgelieferte DSDT ist in der Regel nicht dafür ausgelegt. Und selbst wenn es konzipiert ist, dann für eine Version wie FreeBSD. MacOSX ist ein ernstzunehmendes ACPI-System, d.h. verwendet DSDT maximal, so wie Windows 2001 es verwendet, aber nicht Linux, nicht Windows 98, und nicht Windows 2006 (Vista, auf Windows-Server 2003 aufbauend). Das Richtige ist, eine Nachahmung von Windows 2001 zu machen.

Auch wenn Sie bereits "Darwin" haben, bringen Sie es als "Windows 2001" zum Laufen. **Viele BIOSe haben OSYS = 0x07D2 als Wert, aber nicht 0x2410, da es im Attribut angegeben ist.** Obwohl es Berichte gibt, dass 7D6 oder 7D9 für einige Konfigurationen irgendwie besser sind. Sie müssen sich den Algorithmus ansehen.

Auf meinem Skylake 2017 stellte sich als bester Wert 0x7DA heraus. Daher habe ich eine zusätzliche DSDT FixDarwin7_10000, der ein Windows von 2009 nachahmt, d. h. Windows 7. Ich habe es entsprechend 0x7D9, und es beeinflusst den USB 3.0-Betrieb, aus irgendeinem Grund in dem BIOS wird es für ein WindowsXP-System deaktiviert Auch hier wird das Patch möglicherweise nicht benötigt, wenn diese Kennung im DSDT nicht weiter verwendet wird. Haben Sie das überprüft?

Moment **#3**. Der Hersteller des Motherboards, und damit auch dessen BIOS und dessen DSDT, kann nicht vorhersehen, welche CPU installiert wird, welche Grafikkarte und andere PCI Geräte. Diese sollten aber in der DSDT vorgeschrieben werden! Und umgekehrt, sollten von der DSDT, solche Geräte wie Lautsprecher, Diskettenlaufwerk, Parallelport ausgeschlossen werden. Es gibt keine Treiber für sie und es werden keine benötigt. Außerdem ist es oft notwendig, Anschlüsse/Ports hinzuzufügen oder zu entfernen, um einige Geräte, wie z. B. eine Grafikkarte oder ein SATA-Controller nutzen zu können.

DSDT liegt im BIOS und wird im System im AML-Binärcode verwendet. Es existiert der „IASL-Compiler/Decompiler“, der Codes in eine für Menschen verständliche Sprache übersetzt

DSL. Der menschliche Weg der Bearbeitung ist AML->DSL->Bearbeiten->DSL->AML. Und dann gibt es noch

Punkt **Nr. 4**. Die letzte Kompilierung wird aufgrund von Fehlern unmöglich, Syntax und logischen, die ursprünglich im OEM DSDT vorhanden waren. Bei der Bearbeitung ist es erforderlich, diese ebenfalls zu bearbeiten und zu korrigieren. Nun, zur gleichen Zeit, um die semantischen Fehler, die verursacht wurden, zum Beispiel zu korrigieren, und verhindern, dass der Computer nicht einschlafen oder nicht aufwachen kann. Oder vielleicht gibt es auch neue Geräte, die beschrieben werden sollen.

(Und im Allgemeinen ist es seltsam, aber Kompilieren/Dekompilieren ist nicht strikt invers. Der Betrieb, hin und her ändert die Tabelle, oder auch allgemein, hin geht, zurück nicht - es ist erforderlich zu intervenieren. Aus meiner Sicht bedeutet dies, dass der Decompiler mit Bugs verwendet wurde, solche Programmierer haben daran gearbeitet. Und bei Nichteinhaltung von Normen ist notwendig, dass sie als Warnungen und nicht als Fehler markiert werden. Wenn im ursprünglichen AML etwas schief geht, dann müssen Sie verstehen, dass der Computer funktioniert, also etwas falsch interpretiert werden kann).

Wenn wir all dies getan haben, können wir die korrigierte DSDT in den Bootloader verwenden, indem sie sie in den Ordner /EFI/CLOVER/OEM/xxx/ACPI/patched legen, oder, falls der OEM-Computernamen noch nicht bekannt, in den Ordner /EFI/CLOVER/ACPI/patched, oder **das bootfähige System selbst hat eine eigene Version der DSDT, die im Stammverzeichnis der Systemplatte liegt**.

Woher bekomme ich die Quell-DSDT, die gepatcht werden muss? Es gibt verschiedene Möglichkeiten, sie zu erhalten unter Windows, Linux oder sogar OSX. Wenn Clover es irgendwie geschafft hat zu laufen, bietet sich diese Möglichkeit selbst. Es ist notwendig, das grafische Menü aufzurufen und die Taste F4 zu drücken. Wenn Clover auf einer FAT32-Partition installiert ist, kann es alle OEM ACPI-Tabellen, einschließlich unangetasteter DSDT und FADT sichern.

Seien Sie geduldig, wenn Sie auf einem Flash-Laufwerk speichern und es viele Tabellen gibt, kann der Prozess eine merkliche Zeitspanne in Anspruch nehmen. In der aktuellen Revision extrahiert Clover einen Satz von Tabellen, die zuvor auf andere Weise nicht verfügbar waren, einschließlich AIDA64. Es gibt auch einen Weg, zum Speichern der vorgefertigten DSDT-Version auf der Festplatte. Geben Sie dazu in der Clover-Oberfläche in das Optionsmenü, ändern Sie die DSDT-Maske, verlassen Sie dann das Menü und drücken Sie F5. Clover wird Ihre DSDT speichern, korrigiert auf die aktuelle Maske, mit einem Namen wie DSDT-F597.aml, d.h. gepatcht mit Maske 0xF597. Sie können mehrere Varianten erstellen, um sie später zu vergleichen.

Jetzt können Sie eine DSDT-Datei nehmen und sie bearbeiten... Und für diejenigen, die der Sprache ASL nicht mächtig sind, Clover schlägt vor, einige Korrekturen automatisch durchzuführen. Auch diese Frage beantworte ich gleich.

Frage: "Warum kompiliert DSDT immer noch mit Fehlern, nachdem es von Clover bearbeitet wurde?". Ja, die DSDT ist ein Satz von Beschreibungen und Methoden, von denen wir viele nicht benötigen. Clover berührt sie nicht, auch wenn wir die maximale Maske setzen. Fehler können an diesen Stellen unberührt abgelegt sein, und sie bleiben dort. Das hindert aber nicht daran, dass trotzdem alles funktioniert, denn die DSDT funktioniert nicht als Ganzes, sondern nur als ein Satz von Beschreibungen und Methoden.

Lassen Sie uns die Korrekturen im Detail betrachten. In der Revision 4300+ wurde die Beschreibung der Fixes vereinfacht.

AddDTGP bit(0):

Zur Beschreibung von Geräteeigenschaften, mit Ausnahme der Option DeviceProperties, die oben diskutiert wurden, gibt es eine Variante mit der in DSDT vorgeschriebenen _DSM-Methode.

_DSM - Device Specific Method - eine bekannte Ausprägung dieser Methode, die in MacOSX seit Version 10.5 enthält. Diese Methode ein Array, das das Gerät beschreibt und den Aufruf universelle DTGP-Methode, die für alle Geräte gleich ist.

Diese Korrektur fügt diese Methode einfach hinzu, so dass sie dann auf andere Korrekturen angewendet werden kann. Sie hat keinen eigenständigen Wert. Ich habe den Ratschlag gesehen, eine 0x31-Maske aufzusetzen, sie sagen, die anderen Korrekturen sind nicht erforderlich. Aber dann brauchen Sie auch (1) nicht!

FixDarwin bit(1):

Darwin OS imitiert das Windows XP-System. Viele Schlaf- und Helligkeitsprobleme erwachsen aus einer Fehleinschätzung des Systems.

FixDarwin7 bit(16)

Ähnlich, nur vom Windows 7 System nachgeahmt. Ältere DSDTs haben möglicherweise nicht die Möglichkeit, das für ein solches System zu testen. Sie haben Optionen.

FixShutdown bit(2):

Die Funktion `_PTS` wird um eine Bedingung erweitert: wenn `Argument = 5` (aus) ist, sollten keine anderen Aktionen ausgeführt werden. Seltsam, warum ist das so? Dennoch gibt es immer wieder Beweis für die Wirksamkeit dieses Patches für ASUS-Boards, vielleicht auch für andere. In einigen BIOSen hat die DTDT bereits eine solche Prüfung durchlaufen, in diesem Fall sollte eine solche Korrektur deaktiviert werden, wenn sie in der Konfiguration ist. Mit `SuspendOverride=true`, wird diese Korrektur auf die Argumente 3 und 4 erweitert. Das heißt, das Verlassen zu unterbrechen. Andererseits, wenn `HaltEnabler=true` ist, ist dieser Patch wahrscheinlich nicht nötig, Ich brauche ihn aber.

AddMCHC Bit(3):

Ein solches Gerät der Klasse `0x060000` ist normalerweise in DSDTs nicht vorhanden, aber für einigen Chipsätzen ist dieses Gerät wartungsfähig und muss daher Vorschrift zur korrekten Verteilung des PCI-Bus-Power-Managements haben. Die Frage nach die Notwendigkeit eines Patches wird experimentell entschieden. Mehr Erfahrung, dieses Gerät wurde benötigt auf einem Motherboard mit einem Z77-Chipsatz, da sonst Core-Panic beim ersten Start auftritt. Umgekehrt, auf der G41M-Chipsatz (ICH7) verursacht dieser Fix Panik. Leider lässt sich keine allgemeine Regel erkennen.

FixHPET bit(4):

Wie gesagt, dies ist eine wichtige und notwendige Korrektur. Die minimal erforderliche Maske eines DSDT-Patches sieht also wie `0x0010` aus.

FakeLPC Bit(5):

Ersetzt die `DeviceID` des LPC-Controllers, so dass sich der `AppleLPC-Checksum` an diesen anhängt. Er wird benötigt, wenn der Chipsatz nicht für OSX vorgesehen ist (z. B. ICH9). Allerdings, Die Liste der nativen Intel- und NForce-Chipsätze ist so groß, dass die Notwendigkeit eines solchen Patches sehr selten ist. Das System prüft, ob die `AppleLPC Kext` geladen ist, wenn nicht, wird der Patch benötigt. Allerdings ist auch dies keine Tatsache. Manchmal wird der Kext aus dem Speicher entladen, wenn er nicht benötigt wird, obwohl der Chipsatz unterstützt wird.

FixIPIC-Bit (6):

Entfernt den Interrupt vom IPIC-Gerät. Dieser Fix wirkt sich auf die Funktion der Power-Taste aus (Pop-up-Fenster mit den Optionen `Reset`, `Sleep`, `Shutdown`).

FixSBUS-Bit (7):

Fügt den `SMBusController` in den `device tree` (Gerätebaum) ein und entfernt damit die Warnung über seine Abwesenheit aus dem Systemprotokoll. Und schafft auch die richtige Leistungssteuerung, damit beeinflusst es auch den Ruhezustand.

FixDisplay bit(8):

Erzeugt eine Reihe von Videokarten-Patches. Injiziert Eigenschaften, und Geräte selbst, wenn sie nicht vorhanden sind. Injiziert `FakeID`, wenn bestellt. Fügt benutzerdefinierte Eigenschaften hinzu. Der gleiche Fix fügt ein `HDAU-Gerät` für die Audioausgabe über HDMI hinzu. Wenn `FakeID` gesetzt ist, wird die wird es über die `_DSM`-Methode injiziert. (1974) Patches sind für alle Grafikkarten, nur nicht für Intel-Grafikeinheiten (IGPU). Für eingebettete Intel-Karten ist anders Bit zuständig.

[FIX_INTELGFX_100000](#)

Bei Bedarf wird auch ein `HDMI-Gerät (HDAU)` hinzugefügt.

FixIDE bit(9):

Die AppleIntelPIIXATA-Kext-Panik erschien in System 10.6.1. Zwei Lösungen Probleme - Verwendung des korrigierten Kext oder Fixierung des Geräts im DSDT. Und für von moderneren Systemen? Lassen Sie es sein, wenn es einen solchen Controller gibt.

FixSATA bit(10):

Behebt Probleme mit SATA und beseitigt das Vergilben der Festplattensymbole im System durch Nachahmung von ICH6. Eigentlich eine umstrittene Methode, aber ohne diesen Fix habe ich DVD-Discs, die nicht abspielbar sind, und ein DVD-Laufwerk sollte nicht entfernt werden können, d.h., ersetzen Sie einfach die Icons, ist das keine Option!

Es gibt eine Alternative, die durch Hinzufügen von AppleAHCIport.kext gelöst wird. Siehe das Kapitel zum Patchen der Kexts. Und dementsprechend kann dieser Teil weggelassen werden! (Eines der wenigen Stücke, die ich empfehle es nicht?)

FixFirewire bit(11):

Fügt dem Firewire-Controller die Eigenschaft "fwhub" hinzu, wenn Sie einen haben. Wenn nicht, nichts wird nicht passieren. Sie können es nicht beachten, wenn Sie nicht wissen, ob Sie es setzen müssen oder nicht.

FixUSB bit(12):

Der Versuch, die vielen Probleme mit USB zu lösen. Für den XHCI-Controller, wenn bei Verwendung einer nativen oder gepatchten IOUSBFamily ist dieser DSDT-Patch unverzichtbar. Der Apple-Treiber verwendet speziell ACPI, und die Schreibweise in DSDT muss ist korrekt werden. Es besteht kein Konflikt mit den Strings im DSDT.

FixLAN bit(13):

Injizieren die "eingebauten" Eigenschaft für den Netzwerk-karte (NIC) - notwendig für korrekte Bedienung. Das Kartenmodell wird ebenfalls injected - für die Kosmetik.

FixAirport bit(14):

Ähnlich wie beim LAN wird zusätzlich das Gerät selbst angelegt, wenn es nicht bereits in DSDT vorhanden ist. Bei einigen bekannten Modellen wird die DeviceID ersetzt durch unterstützte. Und der airport (Flughafen) wird ohne weitere Patches aktiviert.

FixHDA bit(15):

Korrigieren der Soundkartenbeschreibung in der DSDT, damit der native Treiber AppleHDA funktioniert. AZAL -> HDEF wird umbenannt, layout-id wird injiziert und Pin-Konfiguration.

FixRTC

Entfernt einen Interrupt vom _RTC-Gerät. Dies ist notwendig, und es ist sehr seltsam, dass jemand ein solches Patch ablehnt. Wenn der ursprüngliche Interrupt noch nicht vorhanden ist, dann ist es in Ordnung, das dieser Patch nicht ausreichen wird. Es kam jedoch die Frage auf, ob die Länge der Region bearbeitet werden muss. An es gab keinen CMOS-Reset, die Länge sollte auf 2 gesetzt werden, aber der Satz erscheint im Kernel-Log wie... nur eine Bank...

Ich weiß nicht, was an dieser Meldung falsch ist, sie kann behoben werden, indem die Länge auf 8 gesetzt wird. Aber in In diesem Fall besteht die Gefahr eines Bios-Resets nach dem Sleep. Daher ist ein zusätzliche der Key zum Lösen eines solchen Tricks eingeführt worden:

```
<key>ACPI</key>
```

```
<dict>
```

```
    <key>DSDT</key>
```