

2.22 SSDT CPU Speedstep

Inhaltsverzeichnis

- [1 Eine SSDT ist teil des ACPI. Die Spezifikation ACPI \(Advanced Configuration and Power Interface\) ist einen offenen Standard, mit dem Betriebssysteme die Erkennung und Konfiguration von Computerhardware durchführen können, um beispielsweise das Energiemanagement durchzuführen. ACPI definiert viele Tabellen, die die Schnittstelle zwischen einem ACPI-kompatiblen Betriebssystem und der System-Firmware darstellen. Dies schließt beispielsweise die Differentiated System Description Table \(DSDT\), Secondary System Description Table \(SSDT\) sowie weiteres mit ein.](#)
- [2 config.plist:](#)
- [3 Advanced:](#)
- [4 ssdtPRGen von Pike R. Alpha:](#)

1 Eine SSDT ist teil des ACPI. Die Spezifikation ACPI (Advanced Configuration and Power Interface) ist einen offenen Standard, mit dem Betriebssysteme die Erkennung und Konfiguration von Computerhardware durchführen können, um beispielsweise das Energiemanagement durchzuführen. ACPI definiert viele Tabellen, die die Schnittstelle zwischen einem ACPI-kompatiblen Betriebssystem und der System-Firmware darstellen. Dies schließt beispielsweise die Differentiated System Description Table (DSDT), Secondary System Description Table (SSDT) sowie weiteres mit ein.

SSDTs werden auch dynamisch während der Computer Laufzeit geladen. Das Script `ssdtprgen.sh` erstellt nur eine SSDT in Bezug auf deine verbaute CPU.

Deshalb eignet sich eine SSDT hervorragend um Powermanagement und Speedstep der CPU zu patchen, falls dieses nicht richtig läuft. Noch dazu gibt es nicht nur eine SSDT in einem Board. Die Anzahl kann locker bis zu 10 Stück sein. Auf Laptops ist es üblich, dass in einer dieser SSDTs auch wichtige Daten zu den verbauten Grafik-Karten (sowohl IntelHDs als auch AMD/Nvidias) drinnen steht. Wir werden aber jetzt eine andere SSDT benutzen, um das Takten unseres Prozessors (Speedstep) zu beeinflussen.

Zuerst solltet ihr also testen ob euer Speedstep überhaupt richtig funktioniert. Dazu könnt ihr beispielsweise das [Intel Power Gadget](#) nutzen und damit beobachten, ob eure CPU richtig taktet. Verändert sich der Takt? Taktet die CPU komplett hoch? (zB bei einem Benchmark) Taktet die CPU auch mal ganz tief? (zB 800 MHz). Falls es hier Probleme gibt, benötigt ihr eine SSDT. Diese ist ganz einfach zu erstellen:

2 config.plist:

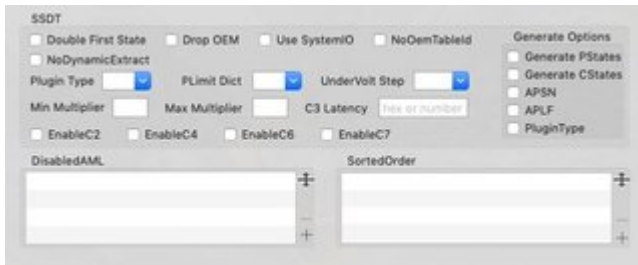
Die einfachste Methode besteht darin, unter Clover den Eintrag PluginType in ACPI --> SSDT zu aktivieren. Daraufhin wird das korrekte CPU Speedstep-Plugin namens X86PlatformPlugin für eure CPU geladen.

Alternativ kann Clover euch nötige P- und C-States für die CPU erstellen.

Das könnt ihr entweder mittels dem Clover Configurator oder einem Textverarbeitungsprogramm machen.

Clover Configurator:

Hierfür einfach nach ACPI-->SSDT navigieren und dort "Generate PStates" und "Generate CStates" aktivieren:



config.plist:

Unter der Sektion `<key>ACPI</key>` --> `<key>SSDT</key>` --> `<key>Generate</key>` --> `<key>Generate</key>` folgendes eintragen:

Code

1. `<key>CStates</key>`
2. `<true/>`
3. `<key>PStates</key>`
4. `<true/>`

3 Advanced:

Erweiterte Methoden und Erklärungen werden hier beschrieben: [CPUFriend Guide, HWP & Speedstep: X86PlatformPlugin vs ACPI_SMC_PlatformPlugin?](#)

4 ssdtPRGen von Pike R. Alpha:

(Die meisten CPUs werden unterstützt. Wenn ihr euch aber nicht sicher seid, schaut hier vorbei: github.com/Piker-Alpha/ssdtPRGen.sh/tree/Beta/Data)

1. Terminal öffnen
2. Script herunterladen hierzu folgenden Befehl im Terminal eingeben Quellcode

Code

1. `curl -o ~/ssdtPRGen.sh https://raw.githubusercontent.com/Piker-Alpha/ssdtPRGen.sh/Beta/ssdtPRGen.sh`

3. Script startfähig machen. Dazu im Terminal eingeben Quellcode

Code

1. `chmod +x ~/ssdtPRGen.sh`

startfähig machen

4. Im Terminal eingeben Quellcode

Code

1. `sudo sh ssdtPRGen.sh`

(hier können weitere Befehle angehängt werden, siehe unten)

5. ENTER drücken

6. Passwort eingeben

7. Frage, ob ihr die SSDT.dsl einsehen wollt beantworten (mit y oder n)

Und dann ist eure SSDT erstmal erstellt. Jetzt müsst ihr sie nur noch einbinden. Dazu geht ihr in den Finder und klickt auf "Gehe zu" und gebt dort ein: `/Benutzer/USERNAME/Library/ssdtPRGen`

In diesem Ordner sollten sich einige SSDTs befinden. Wir brauchen ganz einfach die "SSDT.aml".

Diese kopiert ihr jetzt auf die EFI Partition in den Ordner `/Volumes/EFI/EFI/CLOVER/ACPI/patched/`. Als letztes nur noch einen Neustart machen und die SSDT sollte eingebunden sein. Dadurch sollten sich jetzt auch passende Werte im Intel Power Gadget finden lassen.

Wer zB Overclocking betreibt, muss den Maximal Takt verändern. Wer von Hand sein Prozessor Model

eingeben will, kann dies auch tun. Hier sind die Befehle, die ihr hinter das `sudo sh ssdtPRGen.sh` setzen könnt, um selber weitere Einstellungen zu verändern:

- **-acpi** Prozessor Name (example: CPU0 or C000)
- **-bclk** frequency (base clock frequency, angegeben in Mhz, zB "100")
- **-board-id** (zB: Mac-F60DEB81FF30ACF6)
- **-cpus** number of physical processors [1-4]
- **-debug** output [0/1/3]
- 0 = no debug injection/debug output
- 1 = inject debug statements in: ssdt.dsl
- 2 = show debug output
- 3 = both
- **-extract** ACPI tables nach [Zielpfad]
- **-frequency** (clock frequency)
- **-help** info (Auskunft über alle Befehle)
- **-lfmode**, niedrigste idle Frequenz (zB 800 Mhz)
- **-logical** processors [2-128]
- **-mode** script mode [normal/custom]:
- normal –benutzt ACPI/IOREG Daten des Computers
- custom – benutzt ACPI Daten aus: /Users/Benutzer/Desktop
- **-model** (example: MacPro6,1)
- **-open** die gerade erstellte SSDT
- **-processor** model (zB: 'E3-1285L v3')
- **-show** supported board-id and model combinations:
- Kabylake
- Skylake
- Broadwell
- Haswell
- Ivy Bridge
- Sandy Bridge
- **-target** CPU Typ:
- 0 = Sandy Bridge
- 1 = Ivy Bridge
- 2 = Haswell
- 3 = Broadwell
- 4 = Skylake
- 5 = Kabylake
- **-turbo** maximum (turbo) Frequenz:
- 6300 bei Sandy Bridge und Ivy Bridge
- 8000 für Haswell und Broadwell
- zB 4800 wenn euer Overclock 4,8 Ghz beträgt
- **-tdp** [11.5 - 150]
- **-compatibility** workarounds: [0/1/2/3]
- 0 = keine workarounds
- 1 = inject extra (turbo) P-State am Anfang mit der maximum (turbo) Frequenz + 1 MHz
- 2 = inject extra P-States am Ende
- 3 = beides
- **-xcpm** mode [0/1]
- 0 = XCPM mode disabled
- 1 = XCPM mode enabled