

Bei OpenCore 0.52 CPU Frequency und BusSpeed einstellen

Beitrag von „Franziska1993“ vom 30. Juni 2020, 19:52

Hallo Forum,

ich habe vor kurzem bemerkt, dass mein Ton unter OpenCore 0.52 nicht richtig funktioniert bzw abgehackt klingt

(Music, QuickTime und beim Browser Youtube).

Ein anderer User hatte das gleiche Problem und schrieb:

Das Soundproblem wurde endlich behoben, indem die vollständige Sample-Liste verwendet und die CPU-Frequenz

bearbeitet wurde.



The screenshot shows a table of OpenCore BIOS settings. The table has four columns: Name, Type, Value, and Comment. The rows are: CPU, BusSpeed, OS, FrequencyMhz, TurboBoost, and LoadAllFrequency.

Name	Type	Value	Comment
CPU	Dictionary	Default	
BusSpeed	Number	800000	
OS	Boolean	True	
FrequencyMhz	Number	3000	
TurboBoost	Boolean	False	
LoadAllFrequency	Boolean	False	

Unter Clover wird es hier eingestellt (siehe Screenshot), aber wo und wie in OpenCore.

Ich habe sowohl die OpenCore Doku durchsucht sowie gegoogelt, aber ohne wirklich brauchbaren Ergebniss.

Hat jemand vielleicht einen Vorschlag?

Gruß Franziska1993

Beitrag von „ralf.“ vom 1. Juli 2020, 13:55

Das ist eigentlich hier, PlatformInfo->DataHub->ARTFrequency, FSBFrequency

Hab ich aber noch nicht getestet...

Powermanagement gibt es für den Ryzen @[DSM2](#)

<https://github.com/trulyspinach/SMCAMDProcessor>

Beitrag von „Franziska1993“ vom 1. Juli 2020, 16:32

Hallo ralf.,

ich habe mir mehrere config.plist's angesehen, aber niemand

hat die Werte eingetragen (habe hier bei den "OpenCore Gurus" jedenfalls keine Config-Datei gefunden).

Bei Clover ist die Maske unter CPU eindeutig mit Mhz und Khz.

Wie wird z.B. die CPU-Taktfrequenz von 3600 Ghz in OpenCore eingetragen?

In der Doku wird kein realbezogener Wert als Beispiel aufgeführt.

Gruß Franziska1993

Beitrag von „ralf.“ vom 1. Juli 2020, 19:46

Ich habe das gefunden

<https://github.com/acidanthera/bugtracker/issues/448>

Zitat

[vit9696](#) commented [on 8 Aug 2019](#)

■

I see now the values, and they look crazy. Let me clarify things a little though, because there is a lot of confusion here in addition to code being somewhat wrong as well.

XNU kernel expects us pass up to two parameters `ARTFrequency` and `FSBFrequency`:

- `ARTFrequency` is a platform-dependent constant, which specifies ART timer frequency. By default XNU uses `BASE_ART_CLOCK_SOURCE`, 24 MHz, yet firmware can choose to specify it and override the calculation.
- `FSBFrequency` is bus frequency, that can be derived through the timers and is equivalent to your CPU frequency divided by maximum bus ratio (30 for your setup). This is what we must always specify to XNU.

Notes for OpenCore `OC_CPU_INFO` fields:

- `TSCFrequency` is your CPU frequency calculated from TSC. This probably needs to be renamed to `CPUFrequencyFromTSC`.
- `CPUFrequencyFromART` is missing, and the calculation you pointed out in `OcCpuLib.c` should assign its value to this field.
- `CPUFrequency` is unused, but what it is supposed to contain is resulting CPU frequency XNU will see after multiplying `FSBFrequency` with maximum busratio. The actual value without rounding and measurement errors. This was originally passed for modified kernels (like AMD) to bypass XNU calculation.
- `FSBFrequency` is finally used, and it corresponds to `FSBFrequency` in XNU. The value is calculated from either `CPUFrequencyFromTSC` (legacy) or `CPUFrequencyFromART` (preferred for Skylake) depending on the model.
- `ARTFrequency` should correspond to XNU and contain ART timer frequency, normally `BASE_ART_CLOCK_SOURCE`, yet your CPU is an exception, see below.

So basically there should be an equation:

Code

1. $\text{CPUFrequencyFromART} \approx \text{CPUFrequencyFromTSC} \approx \text{CPUFrequency} \approx \text{FSBFrequency} * \text{MaxBusRatio}$

After the refactoring the names and fixing `CPUFrequency` assignment, for your case it will still be broken, because the calculation uses the wrong values for your CPU. I checked Intel SDM and this is what it says about ART in 17-44 Vol. 3B and 18-128 Vol. 3B:

Code

1. 17.17.4 Invariant Time-Keeper
- 2.
3. The invariant TSC is based on the invariant timekeeping hardware (called
4. Always Running Timer or ART), that runs at the core crystal clock frequency.
5. The ratio defined by CPUID leaf 15H expresses the frequency relationship

6. between the ART hardware and TSC.

7.

Alles anzeigen

Beitrag von „Franziska1993“ vom 1. Juli 2020, 20:02

Hallo ralf.,

danke, so richtig schlau werde ich nicht daraus.

Nur das man an der GHZ-Taktung noch 6 - Nullen dran hängen muss.

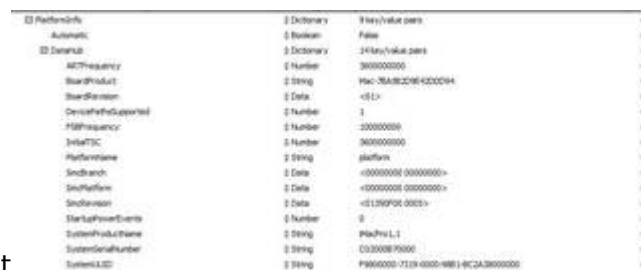
Aus einen anderen Forum habe ich tatsächlich eine config.plist gefunden und folgendes entnommen:

```
<key>FSBFrequency</key>
```

```
<integer>100000000</integer>
```

```
<key>ARTFrequency</key>
```

```
<integer>3600000000</integer>
```



IO PlatformInfo	{ Dictionary	3 key/value pairs	*
Automatic	{ Boolean	False	*
IO DataHub	{ Dictionary	14 key/value pairs	*
ARTFrequency	{ Number	3600000000	*
BoardProduct	{ String	Mac-7BA8C29E-4C2C094	*
BoardRevision	{ Data	<1>	*
CpuStateSupported	{ Number	1	*
FSBFrequency	{ Number	100000000	*
InitRTT	{ Number	360000000	*
PlatformName	{ String	Platform	*
SmuBranch	{ Data	<00000000 00000000>	*
SnpPlatform	{ Data	<00000000 00000000>	*
SnpRevision	{ Data	<1199F04 0000>	*
StartupPowerEvents	{ Number	0	*
SystemProductName	{ String	MacPro1,1	*
SystemSerialNumber	{ String	C013008190000	*
SystemUUID	{ String	F4960000-7119-4000-8B81-4C2A38000000	*

Anbei ein Screenshot

Gruß Franziska1993

Beitrag von „ralf.“ vom 2. Juli 2020, 08:01

Okay, dann passt das ja zu den Angaben in dem Thread von Acidanthera.

FSB war bei dem FX glaube ich 200 - wäre dann

```
<key>FSBFrequency</key>
```

```
<integer>200000000</integer>
```

Gruß

Ralf

Beitrag von „Franziska1993“ vom 2. Juli 2020, 13:01

Danke ralf., der FSB-Wert stimmt mit CPU-Z für den FX überein.

Ich werde nächste Woche einen Test durchführen.

Bleiben die Soundprobleme, hole ich mir den USB-Soundadapter.

Gruß Franziska1993