

AppleHDA scheint nicht zuverlässig zu laden

Beitrag von „Keksfamilie“ vom 7. April 2021, 15:23

[ozw00d](#) Da hast du Recht... was man so online findet, ist CML auf 0x1b,0x0.

Ich bin nach dem gegangen, was mir gfxutil ausgespuckt hatte:

Code

1. \$./gfxutil-1/gfxutil -f HDEF
2. 00:1f.3 8086:06c8 /PCI0@0/HDEF@1F,3 = PciRoot(0x0)/Pci(0x1F,0x3)

Warum weicht das bei mir ab?

[MacPeet](#) Danke für die Erklärung,

einmal Pre-SSDTTime:

```
Device (RTC)
{
  Name _MID, Enabled ("MPPE00P") /* RT Real-Time Clock */ // _MID: Hardware ID
  Name _CRS, ResourceTemplate () // _CRS: Current Resource Settings
  {
    00 (Decoded),
    Sub070, // Range Minimum
    Sub070, // Range Maximum
    Sub070, // Alignment
    Sub070, // Length
  }
  DRNGFlags ()
  {
    00
  }
  Method (_STA, 0, NotSerialized) // _STA: Status
  {
    IF ((_STAR == One))
    {
      Return (Zero)
    }
    ELSE
    {
      Return (Zero)
    }
  }
}

Device (TIMER)
{
  Name _MID, Enabled ("MPPE00P") /* PC-class System Timer */ // _MID: Hardware ID
  Name _CRS, ResourceTemplate () // _CRS: Current Resource Settings
  {
    00 (Decoded),
    Sub070, // Range Minimum
    Sub070, // Range Maximum
    Sub070, // Alignment
    Sub070, // Length
  }
  00 (Decoded),
  Sub070, // Range Minimum
  Sub070, // Range Maximum
  Sub070, // Alignment
  Sub070, // Length
  DRNGFlags ()
  {
    00
  }
}
```

RTC hatte die 8, TIMER die 0. Nach kopieren der Patches in ACPI/Patch sieht das wie folgt aus:

```

Scope (HPET)
{
  Name (_HID, EisaId ("MP0000") /* AT Real-Time Clock */) // _HID: Hardware ID
  Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
  {
    DSD (Decoded,
      0x00, // Range Minimum
      0x00, // Range Maximum
      0x00, // Alignment
      0x00, // Length
    )
    DSD (Flags, 0)
  })
  Method (_STA, 0, NotSerialized) // _STA: Status
  {
    If (EState == One)
    {
      Return (0x0F)
    }
    Else
    {
      Return (Zero)
    }
  }
}

Device (TIMER)
{
  Name (_HID, EisaId ("MP0100") /* PC-class System Timer */) // _HID: Hardware ID
  Name (_CRS, ResourceTemplate () // _CRS: Current Resource Settings
  {
    DSD (Decoded,
      0x00, // Range Minimum
      0x00, // Range Maximum
      0x00, // Alignment
      0x00, // Length
    )
    DSD (Decoded,
      0x00, // Range Minimum
      0x00, // Range Maximum
      0x00, // Alignment
      0x00, // Length
    )
    DSD (Flags, 0)
  })
}
}

```

Die 0 und 8 wurden also erfolgreich aus den Devices entfernt. Wenn ich dich richtig verstanden habe, sollten die jetzt an das HPET Device gebunden werden mithilfe des HPET Patches. Den habe ich also auch in den ACPI Ordner geschoben und in die config.plist hinzugefügt. Da steht als Kommentar // noch dabei "HPET_CRS (Needs _CRS to XCRS Rename)", dieser wurde auch in ACPI/Patches eingetragen. Das HPET Device scheint aber nicht die IRQNoFlags zu übernehmen (auch wenn diese im SSDT-HPET.aml deklariert werden):

```

Scope (_SB.PCI0.LPCB)
{
  Device (HPET)
  {
    Name (_HID, EisaId ("MP0103") /* HPET System Timer */) // _HID: Hardware ID
    Name (_UID, Zero) // _UID: Unique ID
    Name (BUF0, ResourceTemplate ()
    {
      Memory32Fixed (ReadWrite,
        0xFED00000, // Address Base
        0x00004000, // Address Length
        _Y41)
    })
    Method (_STA, 0, NotSerialized) // _STA: Status
    {
      If (HPTE)
      {
        Return (0x0F)
      }
      Return (Zero)
    }
    Method (XCRS, 0, Serialized)
    {
      If (HPTE)
      {
        CreateWordField (BUF0, _SB.PCI0.LPCB.HPET._Y41._BAS, HPTE) // _BAS: Base Address
        HPTE = HPTE /* _HPTE */
      }
      Return (BUF0) /* _SB.PCI0.LPCB.HPET.BUF0 */
    }
  }
}
}

```

Habe mal die Ergebnisse die SSDTTime erzeugt hat angehängt (da ist auch meine Vanilla-DSDT dabei) und mein aktualisiertes EFI. Hab ich da noch was übersehen?