

DSDT Patches

Beitrag von „Schneelöwe“ vom 18. März 2012, 01:03

Hrmpf Heute fällt mir natürlich auf, dass ich über so sekundäre Sachen wie die AppleHDA schon gesprochen habe, aber nicht über sowas wie die KernelPanics wegen der AppleIntelCpuPowermanagmentk.kext, oder wie denn CMOS Reset, geschweige denn über Shutdownprobleme, die einige wohl haben (Ich bisher nicht, aber man weiss ja nie). Also, fangen wir mit dem einfachsten Patch an, dem CMOS-Reset.

Code

1. Device (RTC)
2. {
3. Name (_HID, EisaId ("PNP0B00"))
4. Name (_CRS, ResourceTemplate ())
5. {
6. IO (...)
7. IRQNoFlags ()
8. {8}
9. })
10. }

Für das Problem des Resets nehmen wir uns die Length vor, die muss nämlich 0x02 sein. Also so soll das dann Aussehen:

Code

1. Device (RTC)
2. {
3. Name (_HID, EisaId ("PNP0B00"))
4. Name (_CRS, ResourceTemplate ())
5. {
6. IO (...)
7. IRQNoFlags ()
8. {8}
9. })
10. }

Als nächste kommen wir zu der Problematik der KernelPanics wegen der AppleIntelCpuPowermanagement.kext. Diese entstehen, wenn die RTC (Real Time Clock) und der Timer per IRQ (=Interrupt Request) den Prozessor zum Unterbrechen auffordern. Doch wie lösen wir das Problem? Indem wir die IRQs dem Gerät zuweisen, das auch in einem echten Mac interruptet, dem HPET (High Precision Eventtimer). Was selbstverständlich ist: Wenn ihr in der DSDT kein HPET-Device habt, kann das auch keine IRQs senden, womit der Fix hinfällig wird (Womit ich auch verstanden habe, warum mein allererste Thread hier im Forum Quatsch war, hat ja nur 600 Beiträge gedauert 😄).

Also, aus

Code

```
1. Device (RTC)
2. {
3. Name (_HID, Eisald ("PNP0B00"))
4. Name (_CRS, ResourceTemplate ()
5. {
6. IO (...)
7. IRQNoFlags ()
8. {8}
9. })
10. }
11. Device (TIMR)
12. {
13. Name (_HID, Eisald ("PNP0100"))
14. Name (_CRS, ResourceTemplate ()
15. {
16. IO (...)
17. IRQNoFlags ()
18. {0}
19. })
20. Device (IPIC)
21. {
22. Name (_HID, Eisald ("PNP0000"))
23. Name (_CRS, ResourceTemplate ()
24. {
25. IO (...)
26. IRQNoFlags ()
27. {2}
28. })
29. }
```

Alles anzeigen

entfernen wir die IRQs =

Code

```
1. Device (RTC)
2. {
3. Name (_HID, Eisald ("PNP0B00"))
4. Name (_CRS, ResourceTemplate ()
5. {
```

```

6. IO (...)
7. })
8. }
9. Device (TIMR)
10. {
11. Name (_HID, EisaId ("PNP0100"))
12. Name (_CRS, ResourceTemplate ())
13. {
14. IO (...)
15. })
16. }
17. Device (IPIC)
18. {
19. Name (_HID, EisaId ("PNP0000"))
20. Name (_CRS, ResourceTemplate ())
21. {
22. IO (...)
23. })
24. }

```

Alles anzeigen

Und beim HPET fügen wir sie ein, Daher wird aus

Code

```

1. Device (HPET)
2. {
3. Name (_HID, EisaId ("PNP0103"))
4. Name (_CID, EisaId ("PNP0C01"))
5. Name (BUF0, ResourceTemplate ())
6. {
7. Memory32Fixed (ReadOnly,
8. 0xFED00000, // Address Base
9. 0x00000400, // Address Length
10. _Y14)
11. })
12. Method (_STA, 0, NotSerialized)
13. {
14. If (LGreaterEqual (OSYS, 0x07D1))
15. {
16. If (HPAE)
17. {

```

```

18. Return (0x0F)
19. }
20. }
21. Else
22. {
23. If (HPAE)
24. {
25. Return (0x0B)
26. }
27. }
28. Return (Zero)
29. }
30. Method (_CRS, 0, Serialized)
31. {
32. If (HPAE)
33. {
34. CreateDWordField (BUF0, \_SB.PCI0.LPCB.HPET._Y14._BAS, HPT0)
35. If (LEqual (HPAS, One))
36. {
37. Store (0xFED01000, HPT0)
38. }
39. If (LEqual (HPAS, 0x02))
40. {
41. Store (0xFED02000, HPT0)
42. }
43. If (LEqual (HPAS, 0x03))
44. {
45. Store (0xFED03000, HPT0)
46. }
47. }
48. Return (BUF0)
49. }
50. }

```

Alles anzeigen

das hier:

Code

```

1. Device (HPET)
2. {
3. Name (_HID, Eisald ("PNP0103"))

```

```

4. Name (_CID, Eisald ("PNP0C01"))
5. Name (BUF0, ResourceTemplate ())
6. {
7. IRQNoFlags ()
8. {0}
9. IRQNoFlags ()
10. {8}
11. Memory32Fixed (ReadOnly,
12. 0xFED00000, // Address Base
13. 0x00000400, // Address Length
14. _Y14)
15. })
16. Method (_STA, 0, NotSerialized)
17. {
18. If (LGreaterEqual (OSYS, 0x07D1))
19. {
20. If (HPAE)
21. {
22. Return (0x0F)
23. }
24. }
25. Else
26. {
27. If (HPAE)
28. {
29. Return (0x0B)
30. }
31. }
32. Return (Zero)
33. }
34. Method (_CRS, 0, Serialized)
35. {
36. If (HPAE)
37. {
38. CreateDWordField (BUF0, \_SB.PCI0.LPCB.HPET._Y14._BAS, HPT0)
39. If (LEqual (HPAS, One))
40. {
41. Store (0xFED01000, HPT0)
42. }
43. If (LEqual (HPAS, 0x02))
44. {
45. Store (0xFED02000, HPT0)

```

```
46. }
47. If (LEqual (HPAS, 0x03))
48. {
49. Store (0xFED03000, HPT0)
50. }
51. }
52. Return (BUF0)
53. }
54. }
```

Alles anzeigen

Achja, aus dem IPIC, muss auch der IRQNoFlag eintrag raus 😊 !

Und das beste kommt ja immer zum Schluss. Daher kommt jetzt der Shutdownpatch. Wichtig hier ist diesmal kein Device sondern eine globale Methode. Und diese Methode heisst `_PST`. Unglaublicherweise heisst sie ausgeschrieben "Prepare to Shutdown", also wenn´s daran nicht liegen könnte 😁 .

Also, hier der normalcode meiner DSDT:

Code

```
1. Method (_PTS, 1, NotSerialized)
2. {
3. Store (Zero, P80D)
4. P8XH (Zero, Arg0)
5. Store (Arg0, \_SB.PCI0.LPCB.EC0.SYSC)
6. Store (One, \_SB.PCI0.LPCB.EC0.MUTE)
7. If (LEqual (Arg0, 0x04))
8. {
9. Store (\_SB.PCI0.LPCB.EC0.BNEN, \_SB.PCI0.LPCB.EC0.EBNE)
10. }
11. }
```

Alles anzeigen

Und jetzt der gepatchte:

Code

```
1. Method (_PTS, 1, NotSerialized)
2. {
```

```
3. If (LEqual (Arg0, 0x05)) {}
4. Else
5. {
6. Store (Zero, P80D)
7. P8XH (Zero, Arg0)
8. Store (Arg0, \_SB.PCI0.LPCB.EC0.SYSC)
9. Store (One, \_SB.PCI0.LPCB.EC0.MUTE)
10. If (LEqual (Arg0, 0x04))
11. {
12. Store (\_SB.PCI0.LPCB.EC0.BNEN, \_SB.PCI0.LPCB.EC0.EBNE)
13. }
14. }
15. }
```

Alles anzeigen

Und wie ihr seht, habe ich die gesamte methode in eine Else-If anfrage untergergebracht. Aber dieser Patch soll wohl nicht für Geräte mit einer Intel Grafikkarte funktionieren. So, das war´s für jetzt. Da wir das Problem des CMOS-Resets ja gelöst haben, gucke ich das nächste Mal, was ich bezüglich Sleep machen kann.