

Erledigt

## Dell 7350 DSDT.aml Problem

Beitrag von „kuckkuck“ vom 15. November 2018, 16:17

[grt](#) Eigentlich relativ unwichtig, aber nur zu unser beider Verständnis, ich habe mir das ganze auch mal angeschaut und hätte es wie folgt gelöst:

Das Problem tritt bei GPRW auf. GPRW ist in der SaSsdT selbst nicht definiert, sondern kommt aus der DSDT und sieht wie folgt aus:

Code

```
1. Method (GPRW, 2, NotSerialized)
2. {
3.   Store (Arg0, Index (PRWP, Zero))
4.   Store (ShiftLeft (SS1, One), Local0)
5.   Or (Local0, ShiftLeft (SS2, 0x02), Local0)
6.   Or (Local0, ShiftLeft (SS3, 0x03), Local0)
7.   Or (Local0, ShiftLeft (SS4, 0x04), Local0)
8.   If (And (ShiftLeft (One, Arg1), Local0))
9.   {
10.    Store (Arg1, Index (PRWP, One))
11.  }
12. Else
13. {
14.  ShiftRight (Local0, One, Local0)
15.  FindSetLeftBit (Local0, Index (PRWP, One))
16. }
17.
18. Return (PRWP)
19. }
```

Alles anzeigen

GPRW ist also eigentlich eine Methode, kein Integer Object. In der SaSsdT wird GPRW jedoch als `External (GPRW, IntObj)` definiert. Hieraus entsteht wahrscheinlich das Problem, denn der decompiler interpretiert folglich die `Return (GPRW...` falsch und decodiert sie somit falsch. Dadurch kommt der problematische Code zustande:

Code

1. Method (\_PRW, 0, NotSerialized) // \_PRW: Power Resources for Wake
2. {
3. Return (GPRW)
4. 0x09
5. 0x04
6. }

Was ich jetzt machen würde, wäre `External (GPRW, IntObj)` zu `External (GPRW, MethodObj)` zu korrigieren. Damit hat der compiler dann erstmal kein Problem mehr mit Ganzzahlen nach dem Return, diese müssen jetzt nur noch richtig umformatiert werden. Somit korrigiere ich die GPRW Calls zu:

Code

1. Method (\_PRW, 0, NotSerialized) // \_PRW: Power Resources for Wake
2. {
3. Return (GPRW (0x09, 0x04))
4. }

Die Methode GPRW erhält somit auch seine gewollten Werte beim Aufruf einer entsprechenden \_PRW Methode.

Jetzt habe ich gesehen, wie du das ganze gelöst hast, und frage mich, was die bessere Lösung ist 🤔 Was meinst du?