Erledigt SMBUS verhindert Sleep

Beitrag von „apfelnico“ vom 28. August 2020, 20:37

Sobald SBUS per _DSM-Methode deklariert wird, fällt Sleep flach.

Bei meinem X299 ist noch ein weiteres Device auf der gleichen Adresse vorhanden, welches ich unter macOS nicht haben möchte: "SMBS". Dann habe ich "SBUS" deklariert, die initiale _DSM aber weggelassen. Möglicherweise kannst du diesen Part durch den Bootloader erledigen lassen, über "Device Properties". So sieht der Part innerhalb der SSDT für mein System aus:

Code

```
1. Scope (\_SB.PC00.SMBS)
2. {
3. Method (_STA, 0, NotSerialized) // _STA: Status
4. {
5. If (_OSI ("Darwin"))
6. {
7. Return (Zero)
8. }
9. Else
10. {
11. Return (0x0F)
12. }
13. }
14. }
15.
16. Scope (\_SB.PC00.SBUS)
17. {
18. If (_OSI ("Darwin"))
19. {
20. OperationRegion (SMBP, PCI_Config, 0x40, 0xC0)
21. Field (SMBP, DWordAcc, NoLock, Preserve)
22. {
23. , 2,
24. I2CE, 1
25. }
26.
27. OperationRegion (SMPB, PCI_Config, 0x20, 0x04)
```

```
Field (SMPB, DWordAcc, NoLock, Preserve)
{
, 5,
SBAR, 11
}

OperationRegion (SMBI, SystemIO, (SBAR << 0x05), 0x10)
Field (SMBI, ByteAcc, NoLock, Preserve)
{
HSTS, 8,
Offset (0x02),
HCON, 8,
HCOM, 8,
TXSA, 8,
DAT0, 8,
DAT1, 8,
HBDR, 8,
PECR, 8,
RXSA, 8,
SDAT, 16
}

Method (SSXB, 2, Serialized)
{
If (STRT ())
{
Return (Zero)
}

I2CE = Zero
HSTS = 0xBF
TXSA = Arg0
HCOM = Arg1
HCON = 0x48
If (COMP ())
{
HSTS |= 0xFF
Return (One)
}

Return (Zero)
}

```

```
Method (SRXB, 1, Serialized)
{
If (STRT ())
{
Return (0xFFFF)
}

I2CE = Zero
HSTS = 0xBF
TXSA = (Arg0 | One)
HCON = 0x44
If (COMP ())
{
HSTS |= 0xFF
Return (DAT0) /* \_SB_.PC00.SBUS.DAT0 */
}

Return (0xFFFF)
}

Method (SWRB, 3, Serialized)
{
If (STRT ())
{
Return (Zero)
}

I2CE = Zero
HSTS = 0xBF
TXSA = Arg0
HCOM = Arg1
DAT0 = Arg2
HCON = 0x48
If (COMP ())
{
HSTS |= 0xFF
Return (One)
}

Return (Zero)
}

Method (SRDB, 2, Serialized)
```

```
{
If (STRT ())
{
Return (0xFFFF)
}

I2CE = Zero
HSTS = 0xBF
TXSA = (Arg0 | One)
HCOM = Arg1
HCON = 0x48
If (COMP ())
{
HSTS |= 0xFF
Return (DAT0) /* \_SB_.PC00.SBUS.DAT0 */
}

Return (0xFFFF)
}

Method (SWRW, 3, Serialized)
{
If (STRT ())
{
Return (Zero)
}

I2CE = Zero
HSTS = 0xBF
TXSA = Arg0
HCOM = Arg1
DAT1 = (Arg2 & 0xFF)
DAT0 = ((Arg2 >> 0x08) & 0xFF)
HCON = 0x4C
If (COMP ())
{
HSTS |= 0xFF
Return (One)
}

Return (Zero)
}

```

```
157. Method (SRDW, 2, Serialized)
158. {
159. If (STRT ())
160. {
161. Return (0xFFFF)
162. }
163.
164. I2CE = Zero
165. HSTS = 0xBF
166. TXSA = (Arg0 | One)
167. HCOM = Arg1
168. HCON = 0x4C
169. If (COMP ())
170. {
171. HSTS |= 0xFF
172. Return (((DAT0 << 0x08) | DAT1))
173. }
174.
175. Return (0xFFFFFFFF)
176. }
177.
178. Method (SBLW, 4, Serialized)
179. {
180. If (STRT ())
181. {
182. Return (Zero)
183. }
184.
185. I2CE = Arg3
186. HSTS = 0xBF
187. TXSA = Arg0
188. HCOM = Arg1
189. DAT0 = SizeOf (Arg2)
190. Local1 = Zero
191. HBDR = DerefOf (Arg2 [Zero])
192. HCON = 0x54
193. While ((SizeOf (Arg2) > Local1))
194. {
195. Local0 = 0x4E20
196. While ((!(HSTS & 0x80) && Local0))
197. {
198. Local0--
199. }
```

```
200.
201. If (!Local0)
202. {
203. KILL ()
204. Return (Zero)
205. }
206.
207. Local1++
208. If ((SizeOf (Arg2) > Local1))
209. {
210. HBDR = DerefOf (Arg2 [Local1])
211. HSTS = 0x80
212. }
213. }
214.
215. HSTS = 0x80
216. If (COMP ())
217. {
218. HSTS |= 0xFF
219. Return (One)
220. }
221.
222. Return (Zero)
223. }
224.
225. Method (SBLR, 3, Serialized)
226. {
227. Name (TBUF, Buffer (0x0100){})
228. If (STRT ())
229. {
230. Return (Zero)
231. }
232.
233. I2CE = Arg2
234. HSTS = 0xBF
235. TXSA = (Arg0 | One)
236. HCOM = Arg1
237. HCON = 0x54
238. Local0 = 0x0FA0
239. While ((!(HSTS & 0x80) && Local0))
240. {
241. Local0--
242. Stall (0x32)
```

```
243.  }
244.
245.  If (!Local0)
246.  {
247.  KILL ()
248.  Return (Zero)
249.  }
250.
251.  TBUF [Zero] = DAT0 /* \_SB_.PC00.SBUS.DAT0 */
252.  HSTS = 0x80
253.  Local1 = One
254.  While ((Local1 < DerefOf (TBUF [Zero])))
255.  {
256.  Local0 = 0x0FA0
257.  While ((!(HSTS & 0x80) && Local0))
258.  {
259.  Local0--
260.  Stall (0x32)
261.  }
262.
263.  If (!Local0)
264.  {
265.  KILL ()
266.  Return (Zero)
267.  }
268.
269.  TBUF [Local1] = HBDR /* \_SB_.PC00.SBUS.HBDR */
270.  HSTS = 0x80
271.  Local1++
272.  }
273.
274.  If (COMP ())
275.  {
276.  HSTS |= 0xFF
277.  Return (TBUF) /* \_SB_.PC00.SBUS.SBLR.TBUF */
278.  }
279.
280.  Return (Zero)
281.  }
282.
283.  Method (STRT, 0, Serialized)
284.  {
285.  Local0 = 0xC8
```

```
286. While (Local0)
287. {
288. If ((HSTS & 0x40))
289. {
290. Local0--
291. Sleep (One)
292. If ((Local0 == Zero))
293. {
294. Return (One)
295. }
296. }
297. Else
298. {
299. Local0 = Zero
300. }
301. }
302.
303. Local0 = 0x0FA0
304. While (Local0)
305. {
306. If ((HSTS & One))
307. {
308. Local0--
309. Stall (0x32)
310. If ((Local0 == Zero))
311. {
312. KILL ()
313. }
314. }
315. Else
316. {
317. Return (Zero)
318. }
319. }
320.
321. Return (One)
322. }
323.
324. Method (COMP, 0, Serialized)
325. {
326. Local0 = 0x0FA0
327. While (Local0)
328. {
```

```
If ((HSTS & 0x02))
{
Return (One)
}
Else
{
Local0--
Stall (0x32)
If ((Local0 == Zero))
{
KILL ()
}
}
}

Return (Zero)
}

Method (KILL, 0, Serialized)
{
HCON |= 0x02
HSTS |= 0xFF
}

Device (BUS0)
{
Name (_CID, "smbus") // _CID: Compatible ID
Name (_ADR, Zero) // _ADR: Address
Device (BLC0)
{
Name (_ADR, Zero) // _ADR: Address
Name (_CID, "smbus-blc") // _CID: Compatible ID
If (_OSI ("Darwin"))
{
Method (_DSM, 4, NotSerialized) // _DSM: Device-Specific Method
{
If ((Arg0 == ToUUID ("a0b5b7c6-1318-441c-b0c9-fe695eaf949b")))
{
Local0 = Package (0x02)
{
"refnum",
Zero
}
```

```
372. DTGP (Arg0, Arg1, Arg2, Arg3, RefOf (Local0))
373. Return (Local0)
374. }
375.
376. Return (Zero)
377. }
378. }
379. }
380. }
381.
382. Device (BUS1)
383. {
384. Name (_CID, "smbus") // _CID: Compatible ID
385. Name (_ADR, One) // _ADR: Address
386. }
387. }
388. }
```

Alles anzeigen

Edit:

sobald ich das dazu gepackt hatte, funktioniert Sleep ebenfalls nicht mehr, also "Jacke wie Hose". Nur der Part in der SSDT macht was es soll ...