

ACPI EC: Working around broken hardware

Alexey Starikovskiy

Software Engineer

astarikovskiy@suse.de

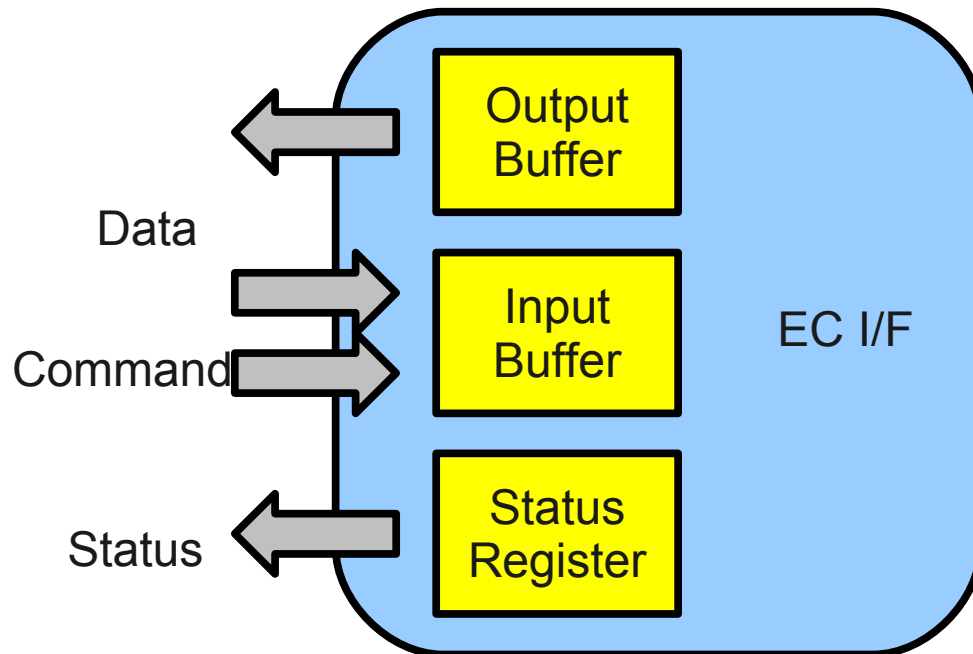
Novell[®]

Background

- Advanced Configuration and Power Interface Specification
 - 10 years of development
 - More than 600 pages
 - Two major implementations:
 - > Microsoft (in Windows),
 - > Intel (everywhere else)
 - No formal test suite
- Embedded Controller (EC)
 - 20 pages - simple

*Even the simplest thing is hard
- Carl von Clausewitz*

Embedded Controller Interface



- EC_SC I/O register
 - Write command
 - Read data
- EC_DATA I/O register
 - Read/Write data
 - 256B address space
- Status bits
 - OBF
 - IBF
 - BURST
 - SCI

Embedded Controller Interface (cont.)

- ACPI reserves one interrupt
- General Purpose Events, each EC reserves one
 - 255 _Qxx events
- Signals for state register bits change:
 - OBF=1
 - IBF=0

History of EC driver

- Polling mode
 - Check for expected change in status register in a loop
 - Simple
 - Slow, consumes CPU cycles for nothing
- Interrupt mode
 - Sleep waiting for signal
 - Fast, CPU can do other tasks while EC is not ready
 - More fragile

Troublemakers

- How to break EC driver
 - Fail to describe EC properly
 - Mess with transactions

Describing EC

- DSDT (required)

```
Device(EC0) {  
    Name(_HID, EISAID("PNP0C09"))  
    Name(_CRS,  
        ResourceTemplate() {  
            IO(Decode16, 0x62, 0x62, 0, 1) // DATA  
            IO(Decode16, 0x66, 0x66, 0, 1) // Command/Status  
        })  
    Name(_GPE, 0x17)
```

- ECDT (optional)

- Binary fields for the same data
- Command/State address before Data address.

Describing EC - failures

- Fail to provide EC DT, when it's required
 - Had to introduce early parse of DS DT
- Swap command and data addresses in EC DT
 - Add option to skip EC DT use
- Come up with *empty* EC DT
 - Skip broken EC DT

Flowcharts and Signals

- “Optimized” Controllers
 - Fujitsu-Siemens Amilo & Clevo D410 : don't send OBF=1
 - Toshiba U305: don't send IBF=0 after address write
 - Workaround: **degrade** to poll mode
- Extended interface
 - MSI uses additional commands to control brightness
 - Workaround: **degrade** to poll mode for these commands
 - Every new machine needs to be checked and added to DMI list

Flowcharts and Signals – GPE Storm

- Manifestation
 - Lost keyboard events
 - Jerky mouse
- Digging deeper
 - Huge ACPI interrupt count
- Working around
 - Degrade to poll mode during transaction
 - Disable EC GPE during transaction
 - Streamline transaction handling in driver

Conclusion

- New hardware support under Linux
 - It may be already supported
 - Just don't walk too far from specification
 - Help us to help you

Unpublished Work of Novell, Inc. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary, and trade secret information of Novell, Inc. Access to this work is restricted to Novell employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. Novell, Inc. makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for Novell products remains at the sole discretion of Novell. Further, Novell, Inc. reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.



Novell.[®]