Debug Methodology Under UEFI

UEFI Fall Plugfest – October 24-27, 2011
Presented by Jack Wang
Phoenix Technologies
Agenda

• Comparing Debug Options
  – UEFI vs Legacy BIOS Debug
• Problems to Solve
• Design Considerations
• Typical Design
• Example Solution
• Q & A
## Comparing Debug Options

<table>
<thead>
<tr>
<th>Feature</th>
<th>Software Solution</th>
<th>Hardware Solution (ICE)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interface</strong></td>
<td>Simple</td>
<td>Complicated</td>
</tr>
<tr>
<td><strong>Capability</strong></td>
<td>Limited</td>
<td>Strong</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>Platform Independent</td>
<td>Platform Dependent</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Low</td>
<td>Extremely High</td>
</tr>
<tr>
<td><strong>Connection</strong></td>
<td>Universal</td>
<td>Limited</td>
</tr>
</tbody>
</table>

**Software Solution**
- Universal connection
- Low cost
- Platform independent

**Hardware Solution (ICE)**
- Limited connection
- Extremely high cost
- Platform dependent
# UEFI vs Legacy BIOS Debug

<table>
<thead>
<tr>
<th></th>
<th>Nowadays (UEFI)</th>
<th>Prior Art (Legacy BIOS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Language</strong></td>
<td>C Base</td>
<td>Assembly Base</td>
</tr>
<tr>
<td><strong>Debugging</strong></td>
<td>Source Level</td>
<td>Symbolic</td>
</tr>
<tr>
<td><strong>Status Output</strong></td>
<td>Message Driven</td>
<td>POST Code Driven</td>
</tr>
<tr>
<td><strong>Status Interface</strong></td>
<td>ReportStatusCode Device (e.g. Port 80+81)</td>
<td>Port 80</td>
</tr>
<tr>
<td><strong>Debug Interface</strong></td>
<td>USB/I2C/SPI/Wireless</td>
<td>Serial/Parallel/PCI Slot</td>
</tr>
</tbody>
</table>
Problems to Solve

• Legacy Debug Interfaces Are Going Away
  – Serial Port, Parallel Port, PCI Slot, etc.

• Board Design Issues
  – Lack of J-TAG/XDP/USB 2.0 Debug Port

• Reliability
  – Complicated Data Communication

• Conflixtion
  – Debug Interface Not Always Available
Device Considerations

• Cross-Platform (Intel/AMD/ARM...)
• Multiple Connection Methods
• Usability
  – POST Time & Runtime
  – Use in R&D or After Market
  – Use on CRB or Production Hardware
• Status Reporting
  – I/O Port 80/81 (Checkpoints)
  – Debug Messages (Log)
Feature Consideration

• Multiple Phase Support
  – PEI/DXE/SMM/CSM/Runtime
  – Selectable Debug Phase (PEI/DXE/SMM/CSM)
• Pre-RAM Debugging
• Convenience
  – Automatic Source Level Detection
  – Debug Mode Auto-Enabling
  – Debug Interface Auto-Detection
  – Debug Driver Auto-Download
  – Debug/Release Build Override
• Target-Defined Debug Features
Typical Design (Phoenix PDD)

Multiple Interfaces

- Platform End
  - USB
  - I2C (DDR2/3), GPIO
  - Serial
  - SPI
  - PS2

- Host End
  - USB
  - Serial

Stand-alone (Host Free)

- Port 80 Redirection
- LCD Message Panel (Optional)
- Wireless Module (Optional)
Example Solution

- LCD
- USB
- COM
- SPI
- I2C
- Port 80/81
- Patent Pending
Usage Scenarios

• Remote Debugging (Host <-> Target)

• Standalone Debugging (Host Free)
Agenda

• Comparing Debug Options
  – UEFI vs Legacy BIOS Debug
• Problems to Solve
• Design Considerations
• Typical Design
• Example Solution
• Q & A
Thanks for attending the UEFI Fall Plugfest 2011

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

phoenix®

Innovation at the CORE
And that’s all for now ...

Download presentations after the plugfest at www.uefi.org