Best Technical Methods for Unified Extensible Firmware Interface (UEFI) Development

Reducing Platform Boot Times

Michael A. Rothman, Intel

Firmware Debugging: UEFI and USB for Platform Forensics

Brian Richardson, AMI

EFIS003

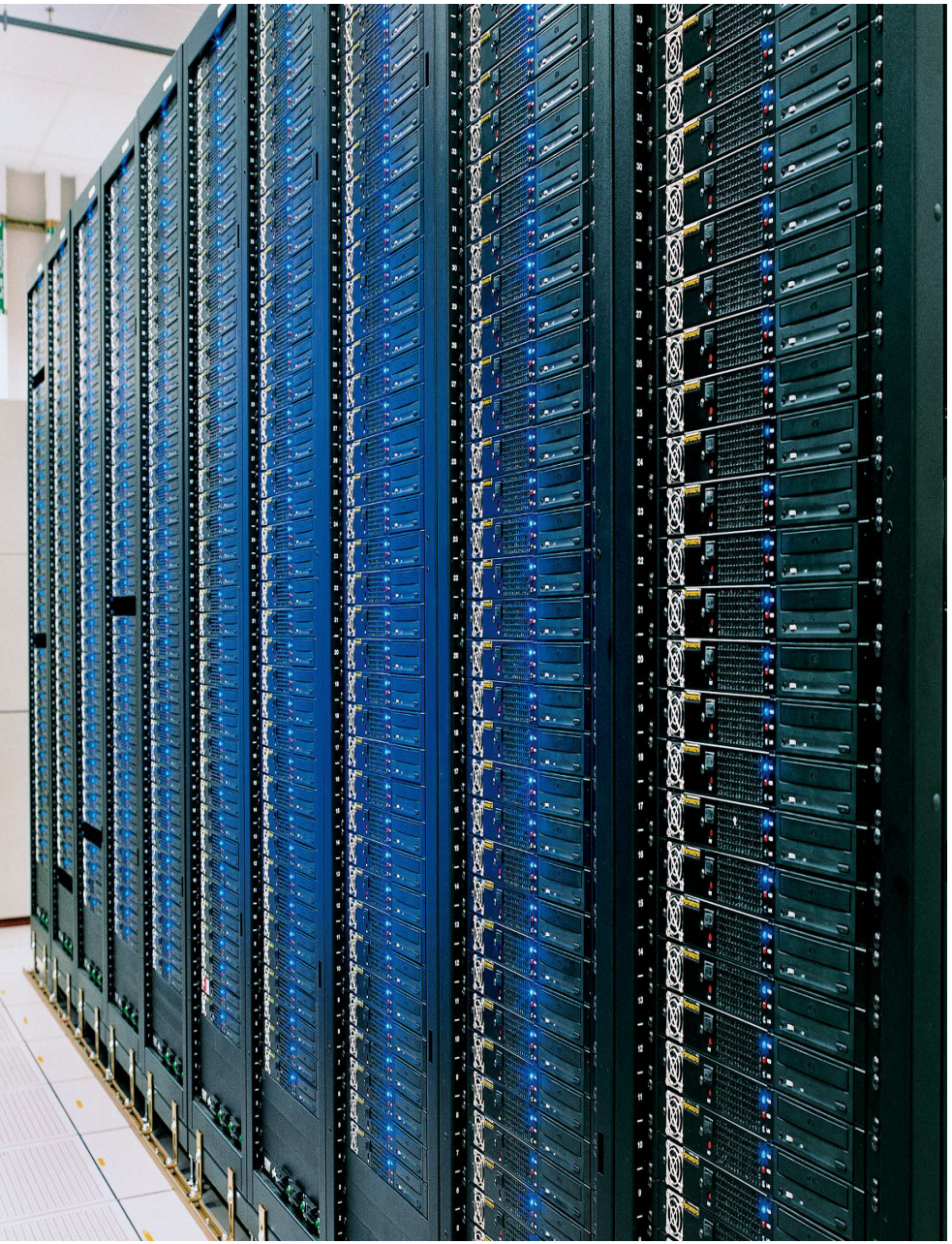
Sponsors of Tomorrow.™ intel

Reducing Platform Boot Times

Michael A. Rothman
Senior Staff Software Engineer

EFIS003

Sponsors of Tomorrow.™ intel



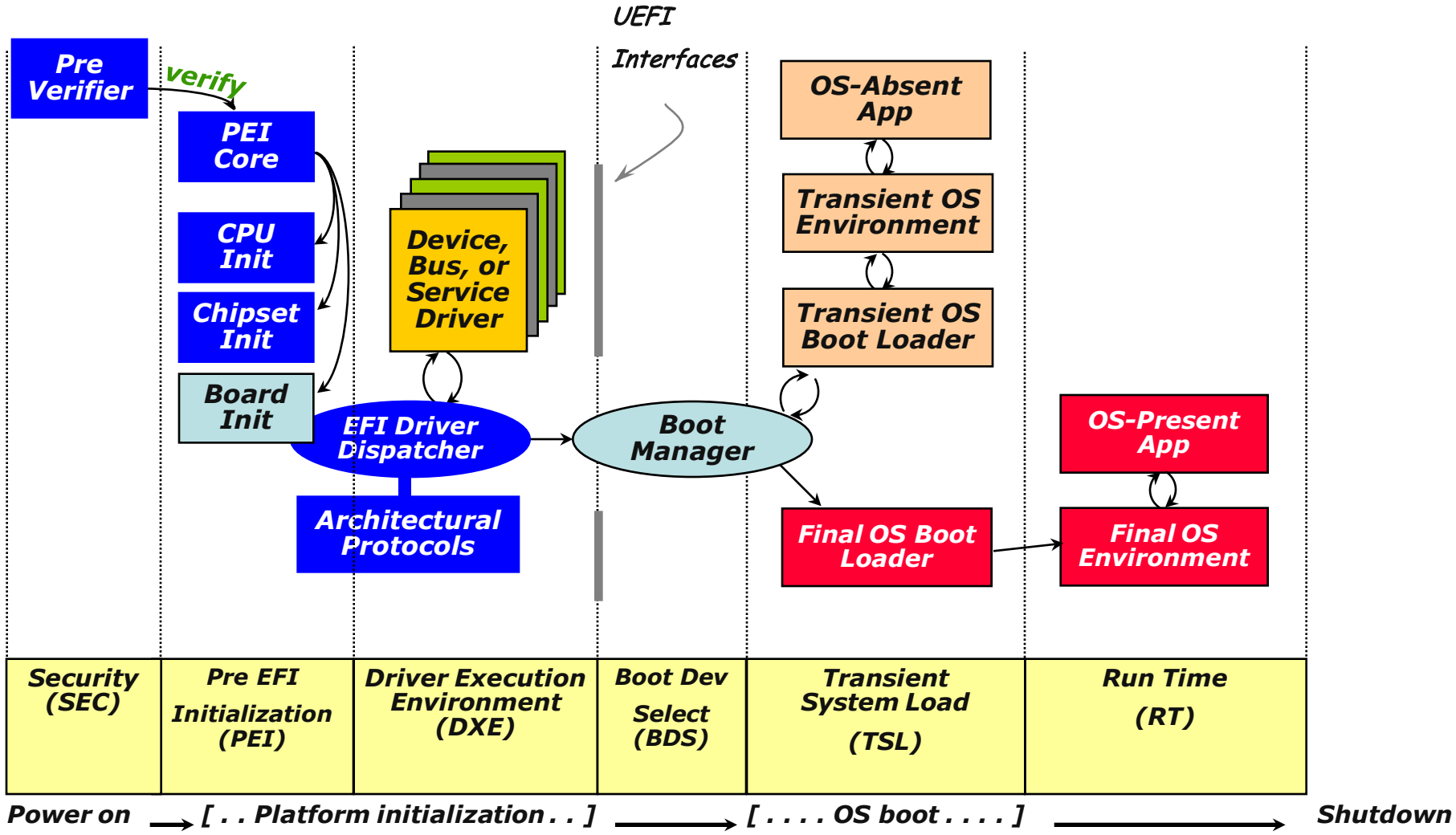
Background Information

Factors in performance

Things to note

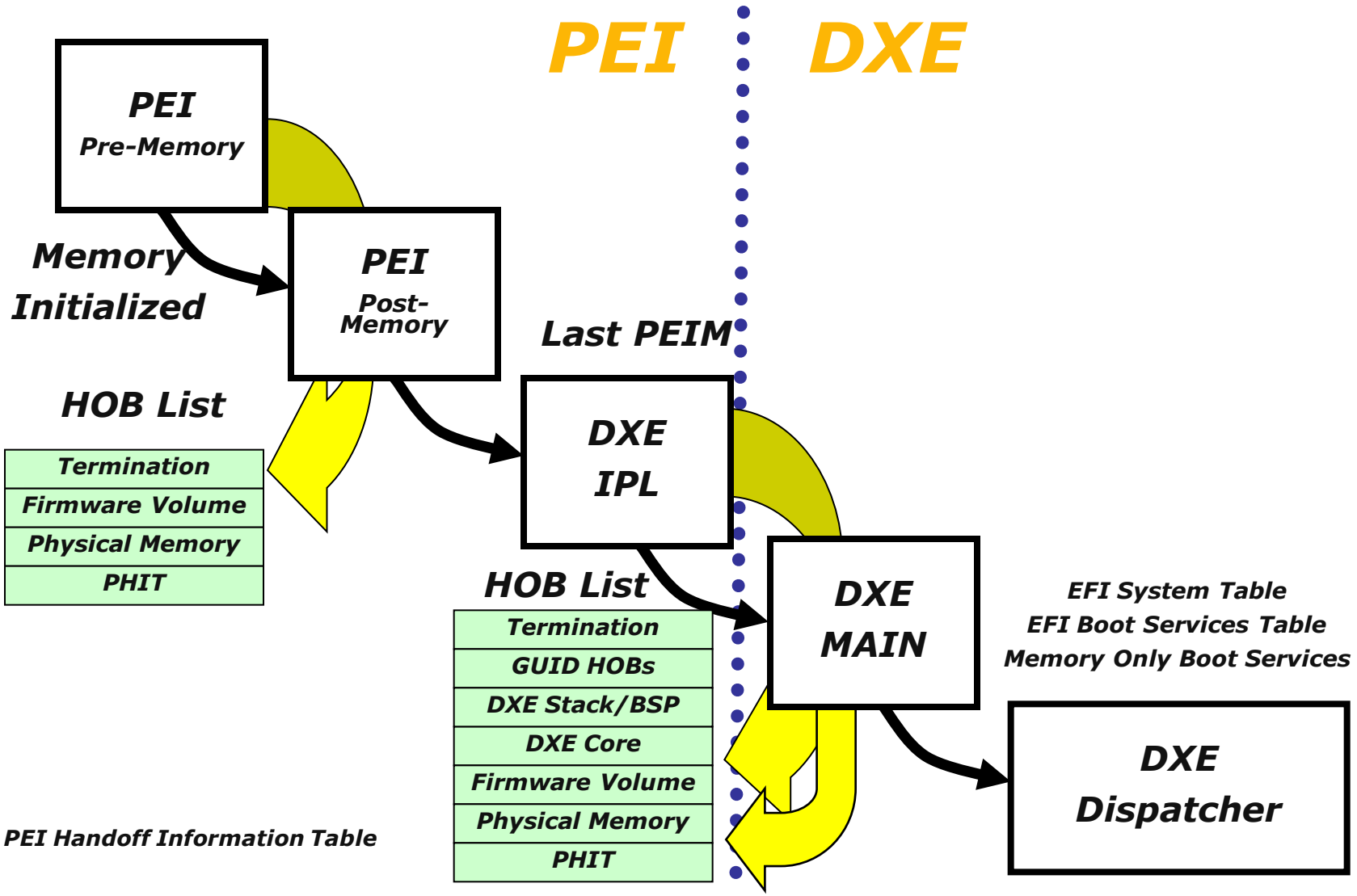
Key Learnings

UEFI Phase Transitions



*

PEI to DXE Transition

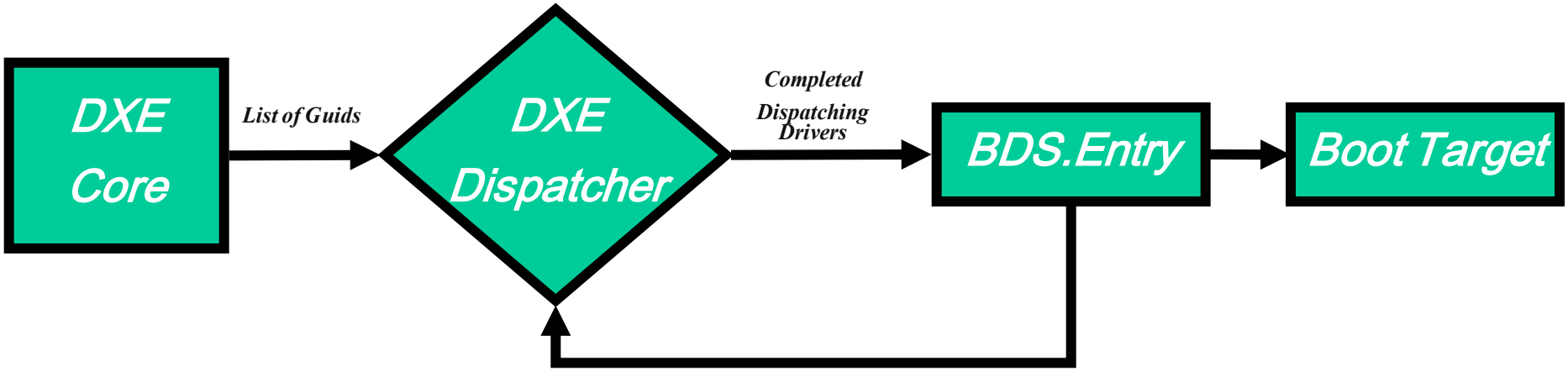


PHIT – PEI Handoff Information Table

*

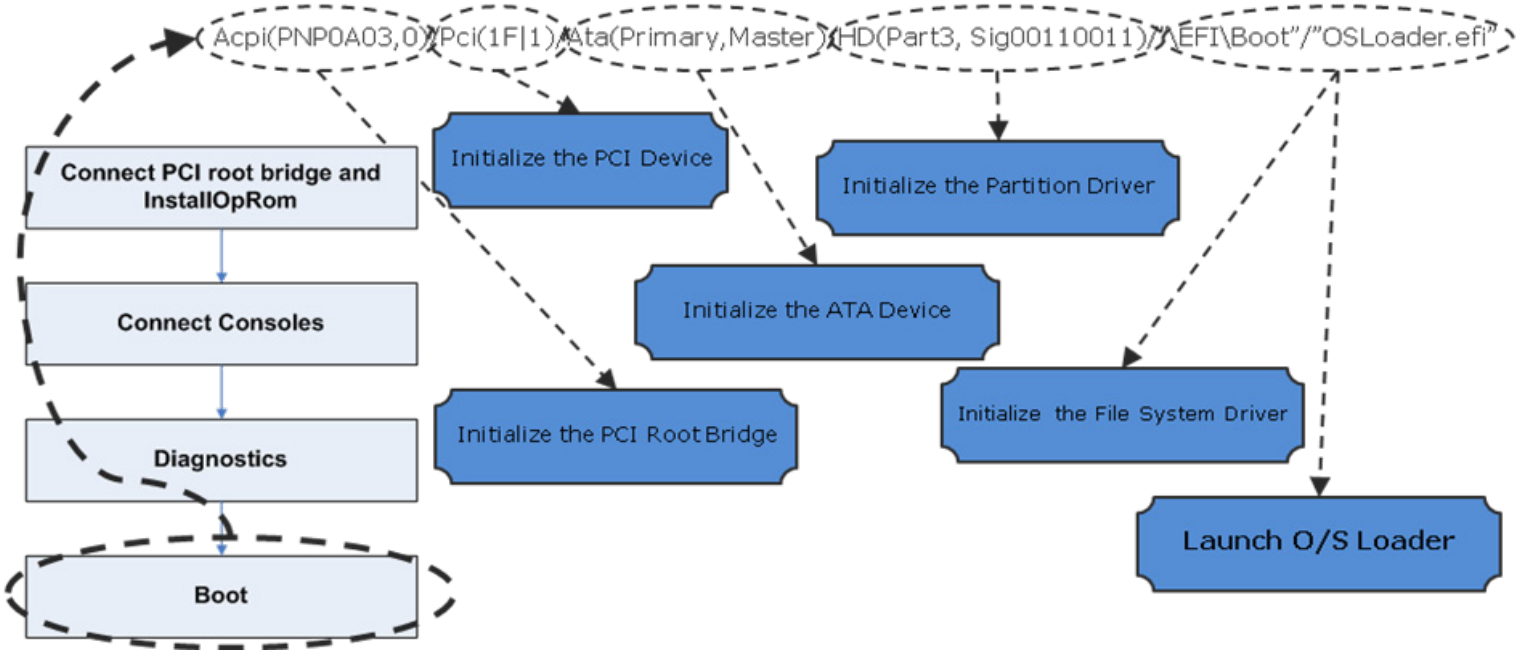
Boot Device Selection

- Invoked after DXE Dispatcher is Complete
- Implemented as a driver
- Connects EFI drivers as required
  - Establishes Consoles (Keyboard, Video)
  - Processes EFI Boot Options (Boots O/S)

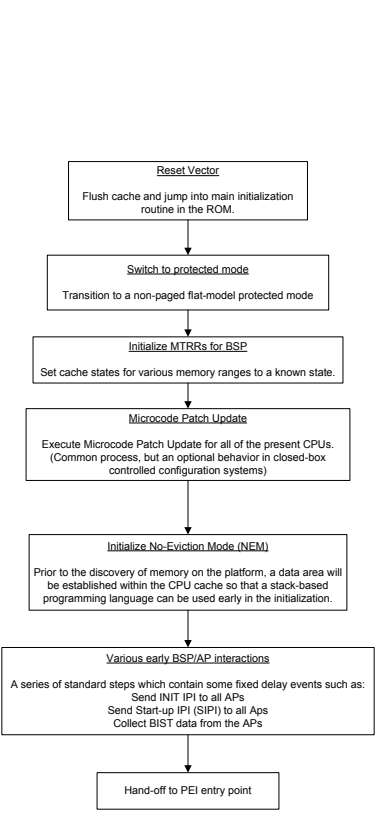


What is a Boot Target?

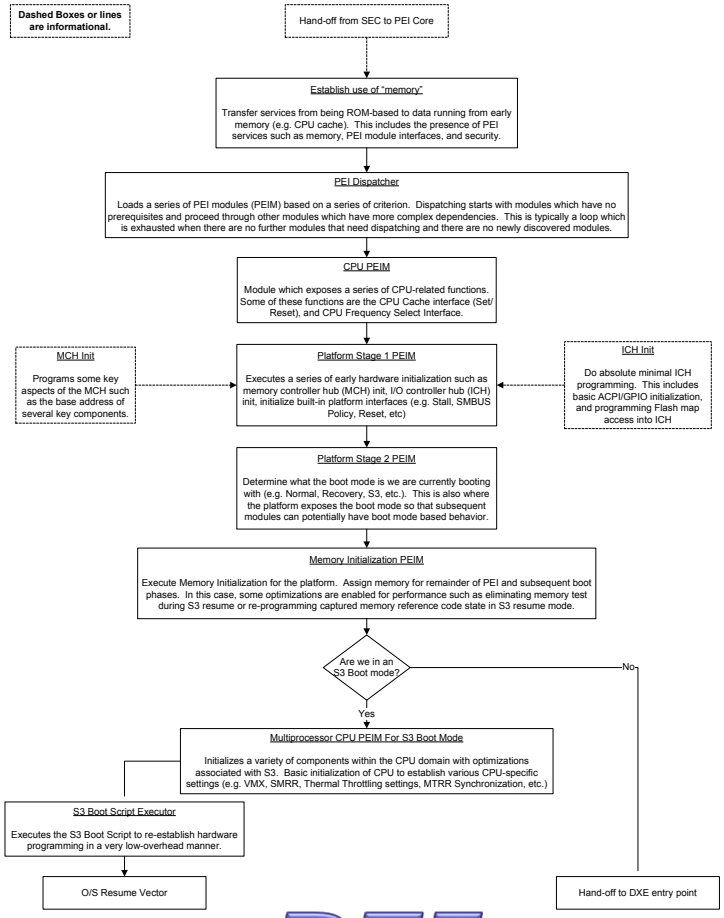
- A boot target is described through an EFI Device Path.
  - A binary description of the physical location of a particular target.



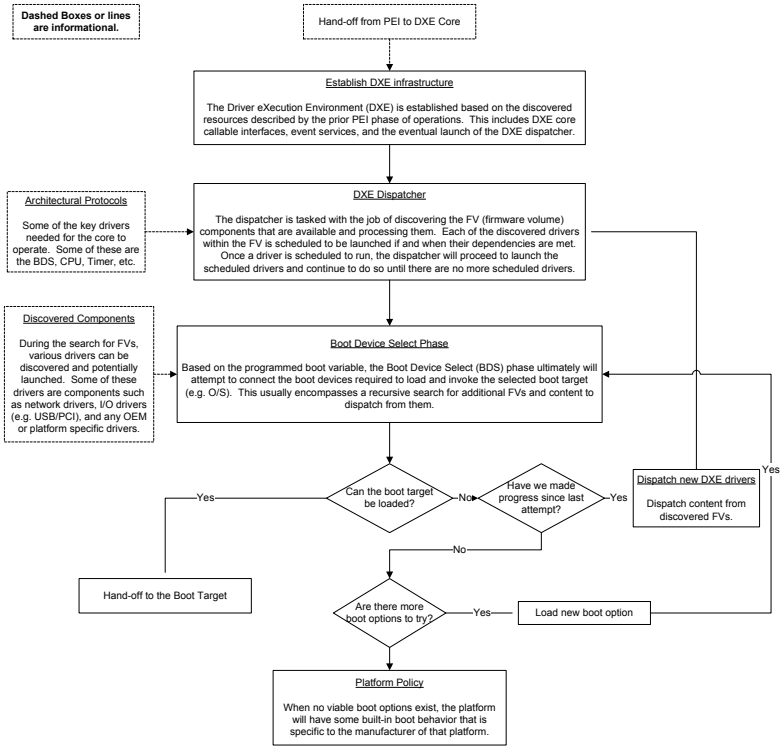
For those who want more detail……



SEC

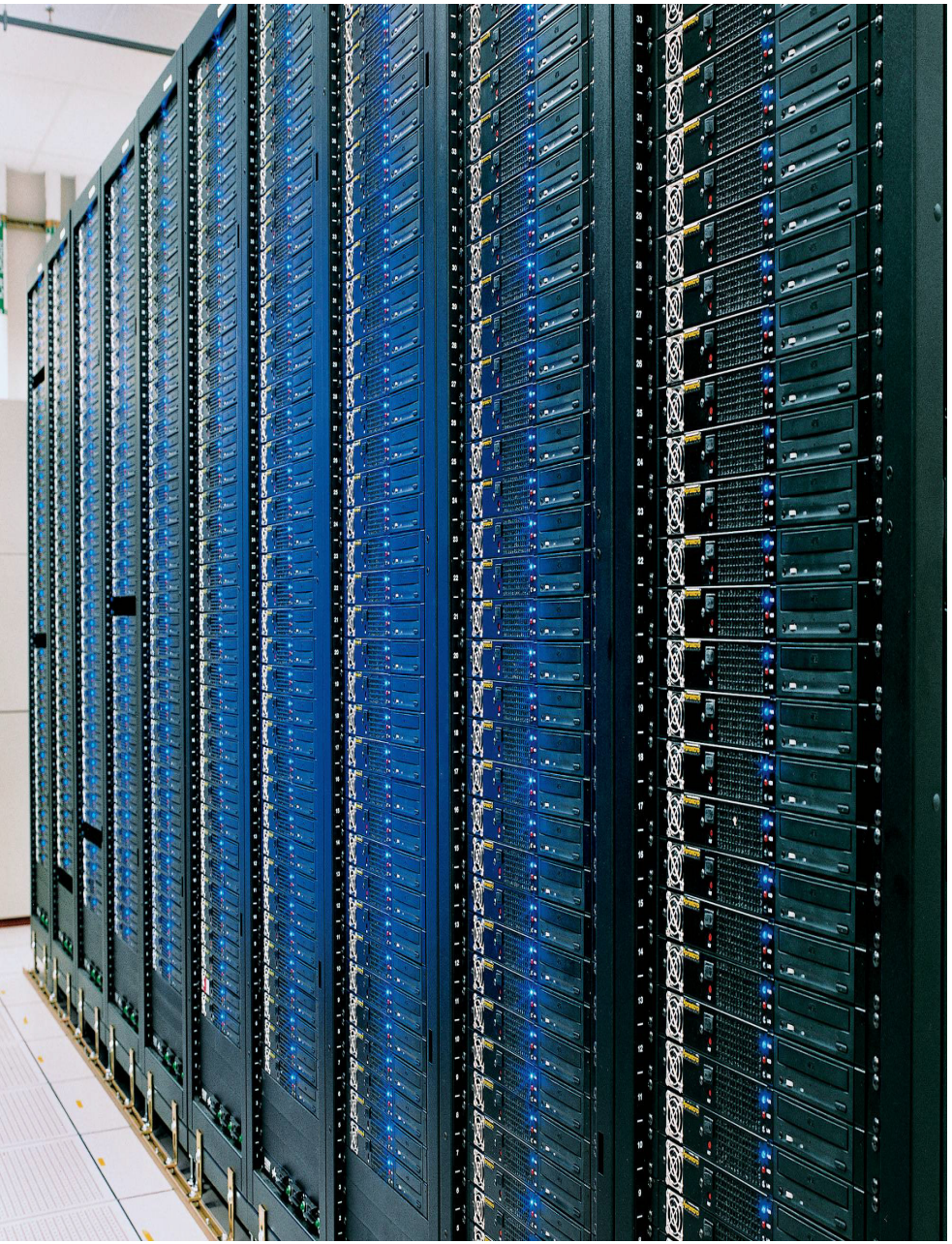


PEI



DXE/BDS

More details in a whitepaper located at: http://edc.intel.com/Link.aspx?id=2355



Background Information

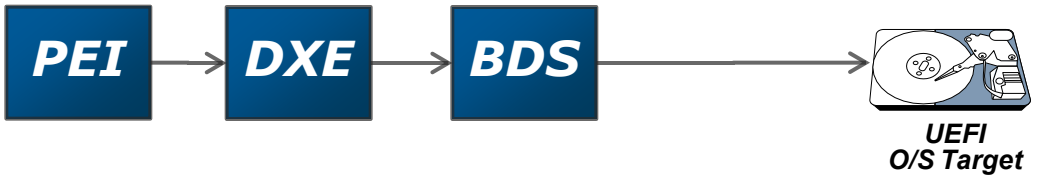
Factors in performance

Things to note

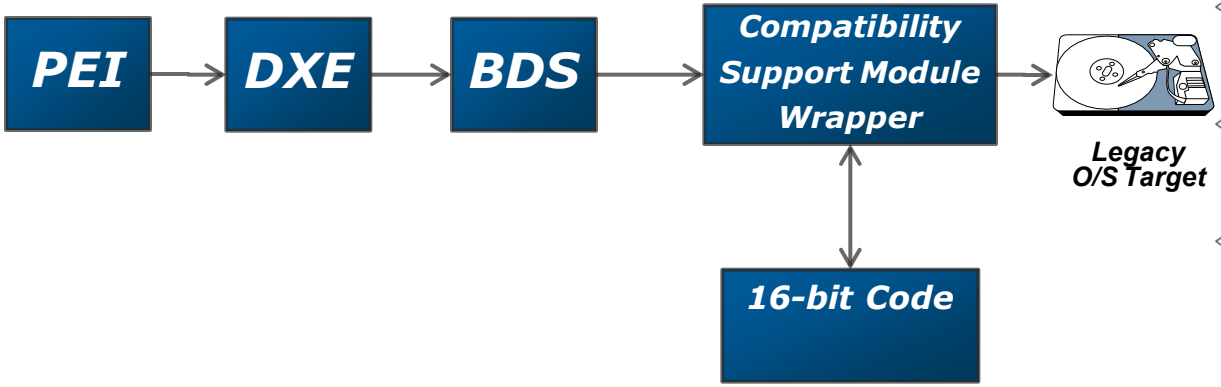
Key Learnings

O/S Target and Attributes

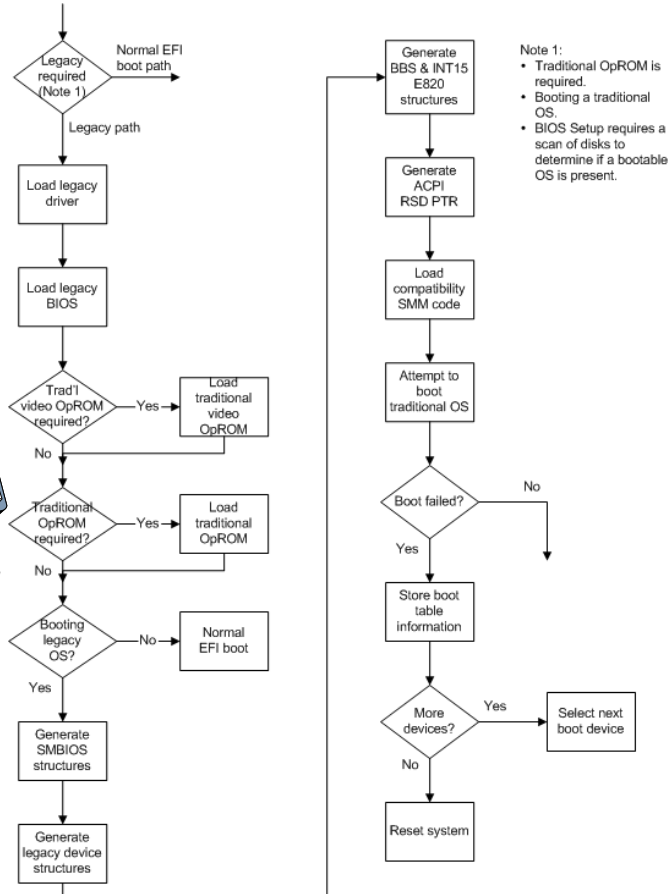
- What are the target Operating Systems?
  - Legacy Boot Support Required?
  - What data does the O/S require from the BIOS?
    - Some tables may not be required for certain targets.



UEFI
O/S Target



Legacy
O/S Target



*

Platform Specific Expectations/Behavior

- What are the platform policies?
  - Expect to interact with user during the pre-boot?

Interactive Boot

No pre-boot interaction

  - What type of hardware are we required to initialize prior to launching the O/S?

Platform Policy choices affect boot times

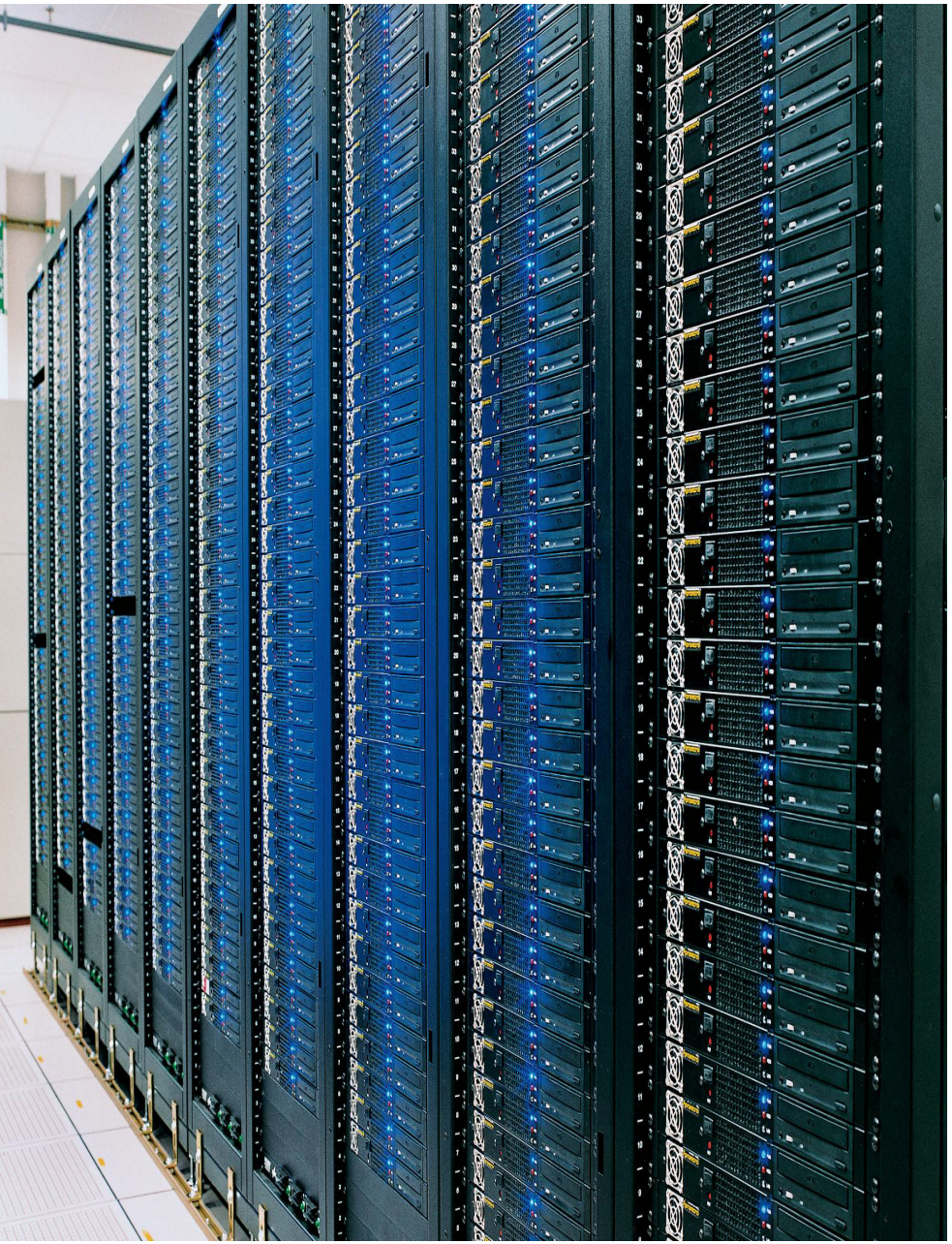
Peripherals Affect Performance

- Can we avoid slow hardware?
  - Use of an SSD boot device in lieu of rotating media can save seconds in the boot time.

Values	DRAM	SSD (34nm)	EIDE
Read Latency	~30 ns	65 µs	8.5 ms
Read BW (MB/s)	1800	250	120
Write Latency	~30 ns	85 µs	10 ms
Write BW (MB/s)	1800	70	120
Spin-up/down time	N/A	N/A	1-2s++

Higher the RPM longer the time

Boot hardware can affect times tremendously



Background Information

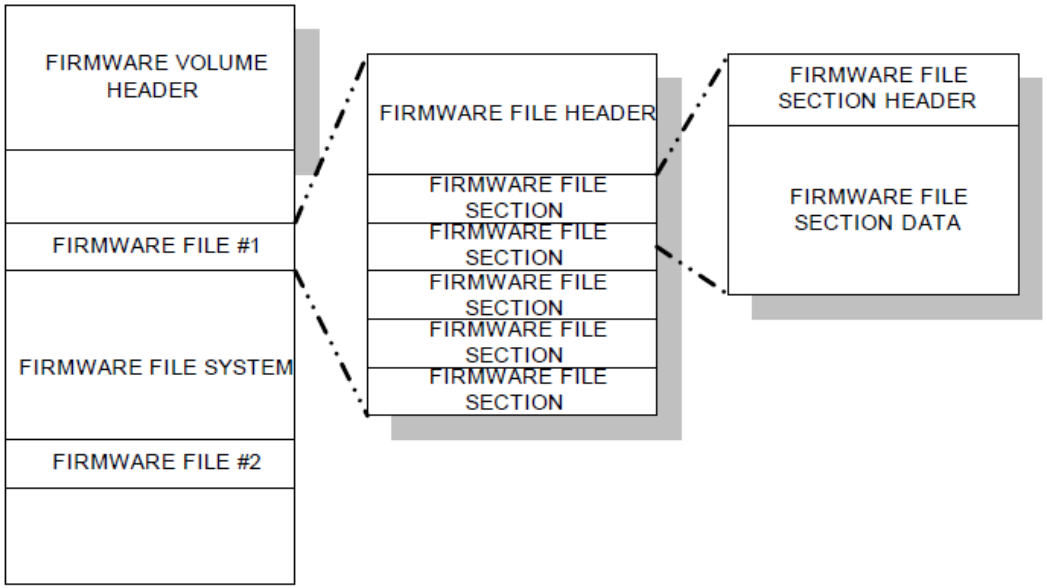
Factors in performance

Things to note

Key Learnings

Size and Organization matters!

- The less you have to read from FLASH the better.
  - It is possible to organize the FLASH layout so that you never search firmware volumes which contain nothing of interest for that configuration.



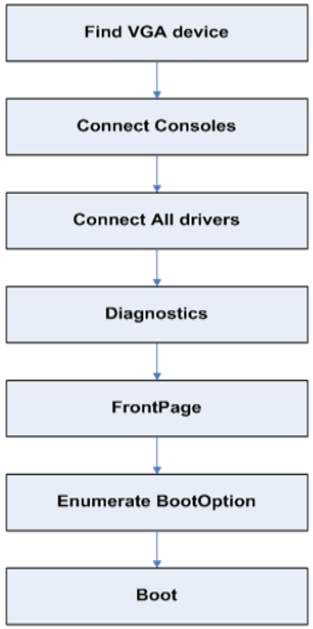
Where to Optimize?

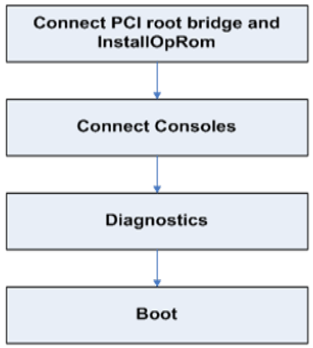
- Try to avoid slowing down the boot process for to accommodate the case which almost never happens.
  - Pausing for a keystroke in the anticipation that someone might interrupt the boot process.
  - Initializing and reading from alternate recovery devices when in almost all cases, we aren't going to be asked to recover the platform.

Platform behavior requirements often dictate where certain optimizations can occur.

Functional Optimization

- Note that depending on platform needs, we may very well do different things….



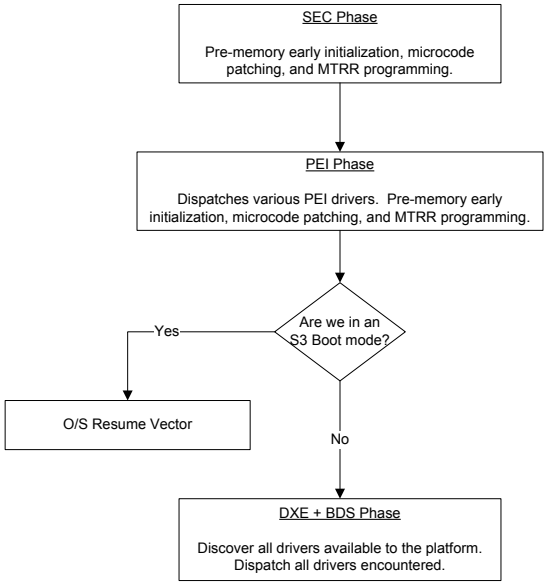


For a normal boot, the figure on the **left** illustrates a common set of operations during the boot. The figure **above** shows an optimized boot. Both are accomplishing the same basic goal – launching the boot target

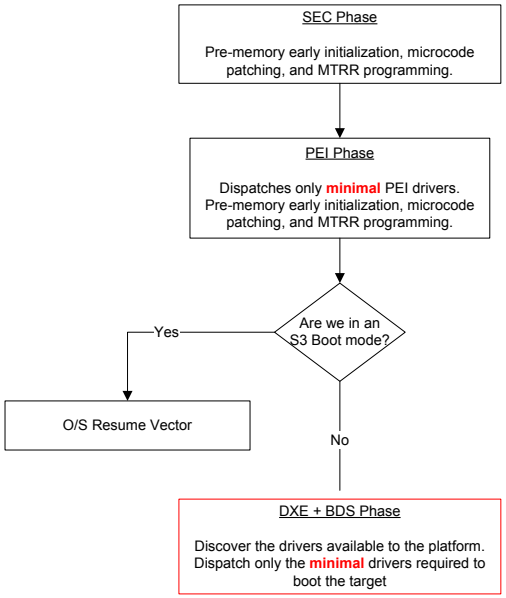
BIOS functionality can and will vary

Maintain Architectural Design

- Performance Optimization doesn't mean we lose UEFI compatibility

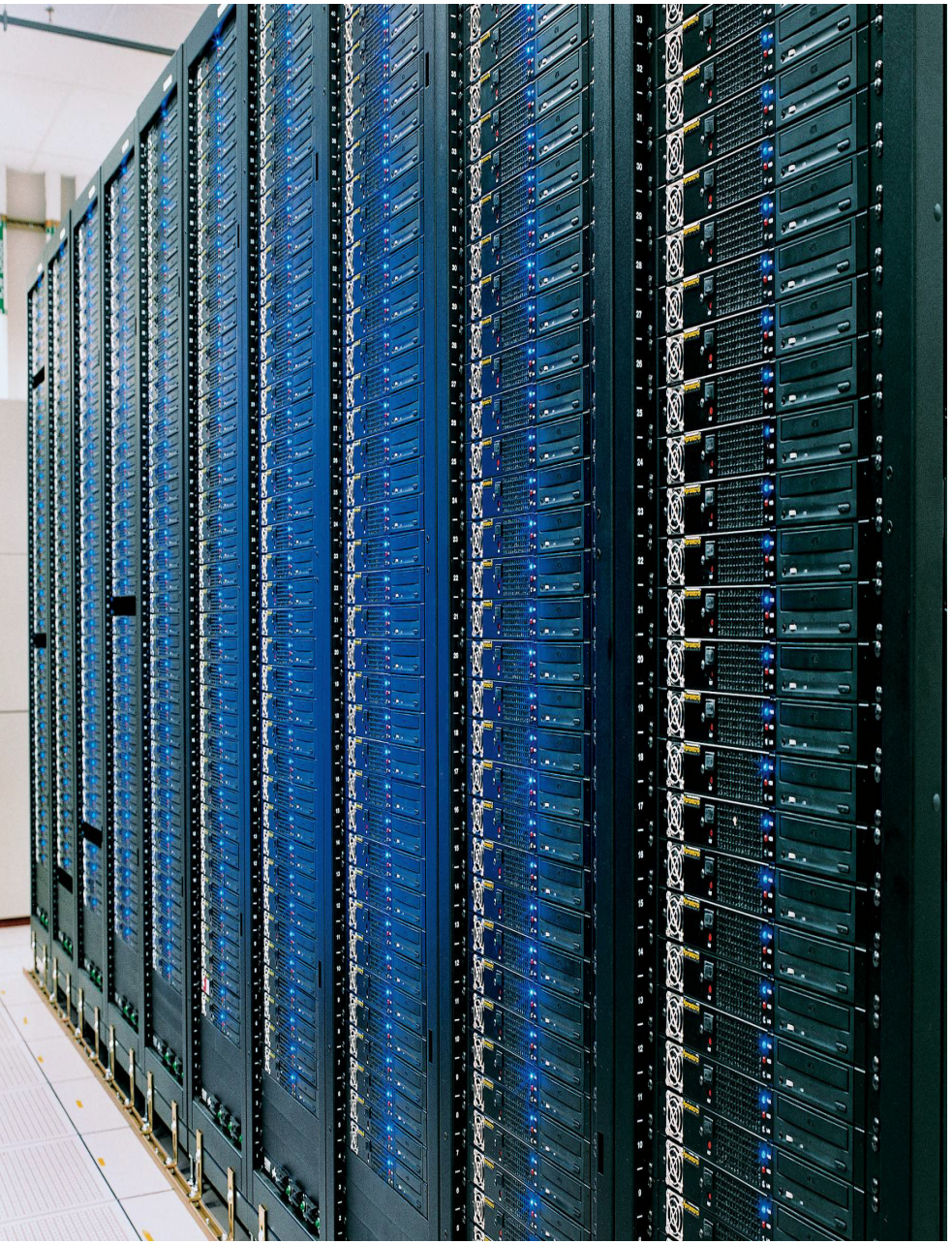


Normal Boot



Optimized Boot

Optimize without losing UEFI compatibility



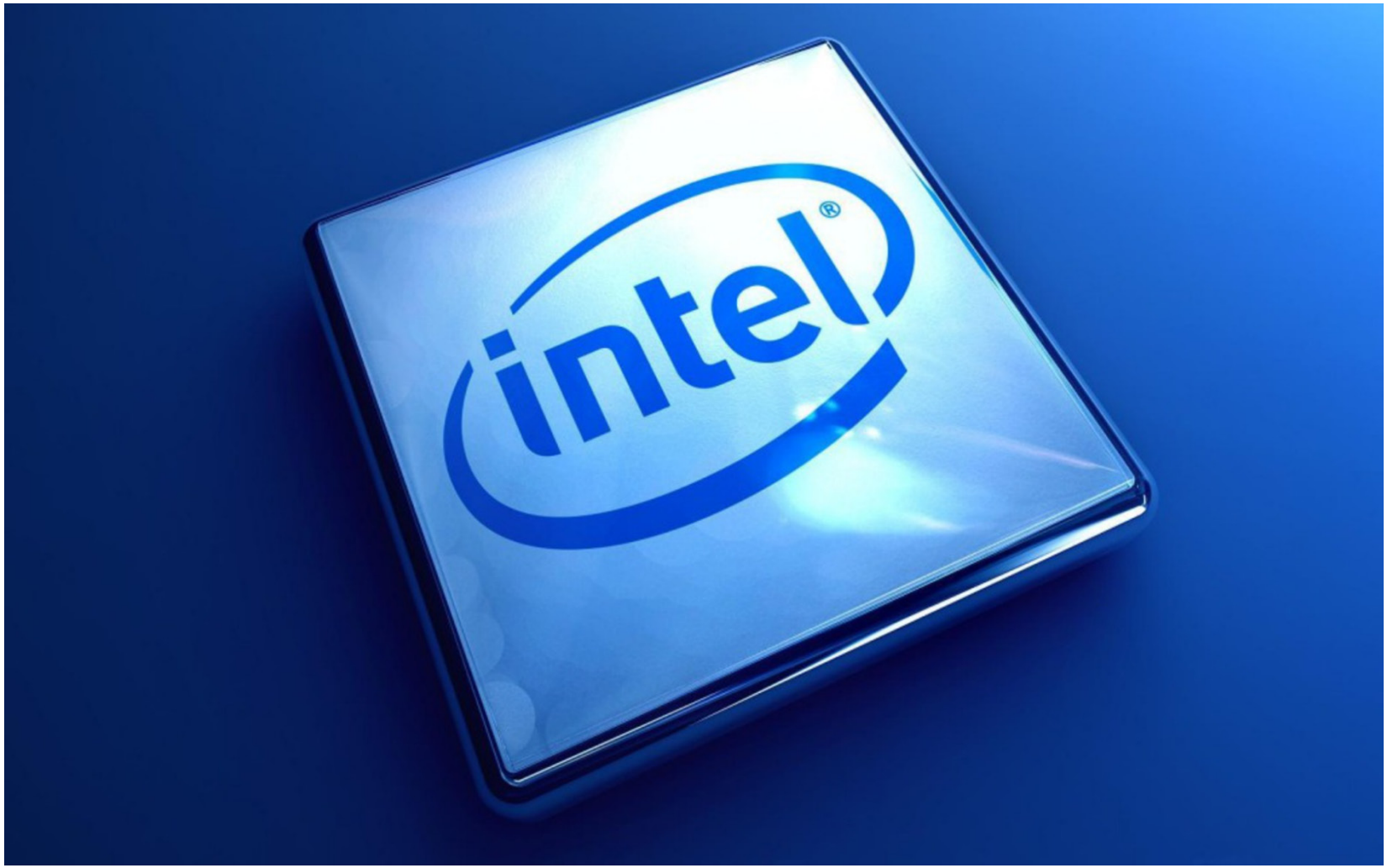
Background Information

Factors in performance

Things to note

Key Learnings

Demo Video




```
SEC    Phase Duration :      26342  (us)
PEI    Phase Duration :    1230905  (us)
DXE    Phase Duration :     998234  (us)
BDS    Phase Duration :    7396050  (us)
Total         Duration :  9.651531  (s)
```

Normal Boot

```
SEC    Phase Duration :      26419  (us)
PEI    Phase Duration :     763315  (us)
DXE    Phase Duration :     443021  (us)
BDS    Phase Duration :     766778  (us)
Total         Duration :  1.999533  (s)
```

Optimized Boot

Key Learnings

- Performance can be greatly affected by Platform Policy and Hardware Configurations
  - Firmware engineers get involved early in the platform design
- BIOS Design Elements Can Improve Performance
  - A variety of software optimization techniques exist within the BIOS
- Performance Optimization does not mean a lack of compatibility
- See the published whitepaper for more details: http://edc.intel.com/Link.aspx?id=2355

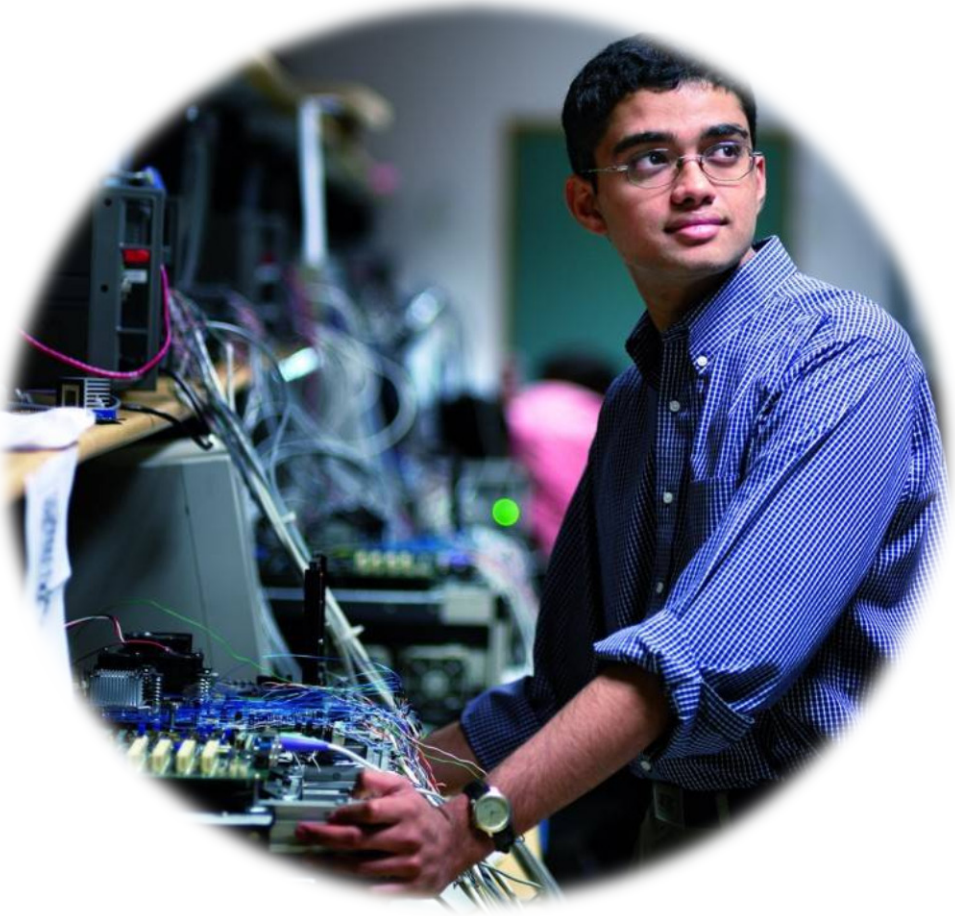
Firmware Debugging: UEFI and USB for Platform Forensics

Brian Richardson - American Megatrends, Inc.
Senior Technical Marketing Engineer

American Megatrends

Sponsors of Tomorrow.™ intel

Agenda



Limitations for UEFI Debugging

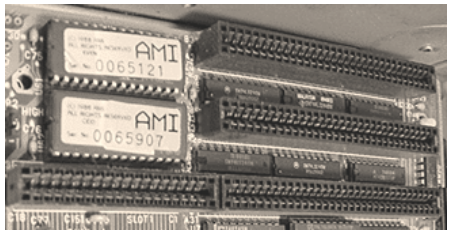
Utilizing USB Debug Solutions

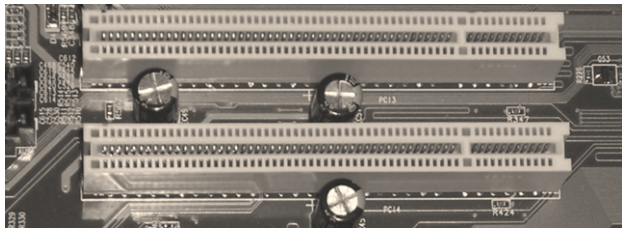
Extending UEFI Debugging Concepts

Using USB Debugging in the Field

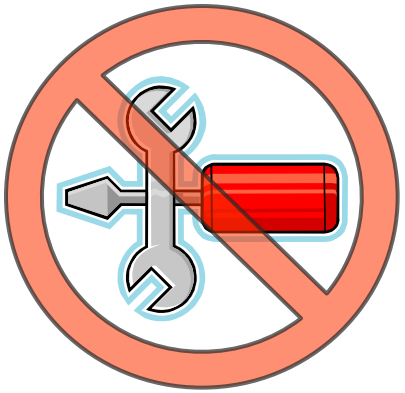
Limitations for UEFI Debugging

- Moving to UEFI introduced new debug tools
  - Debug Strings, Status Codes, C-style debugging
  - Problem: these tools are for developers, not users
- Tools from "the BIOS days" are disappearing





- "No user-serviceable parts inside"
  - Thin & light systems
  - Netbook, nettop, embedded
  - No expansion slots



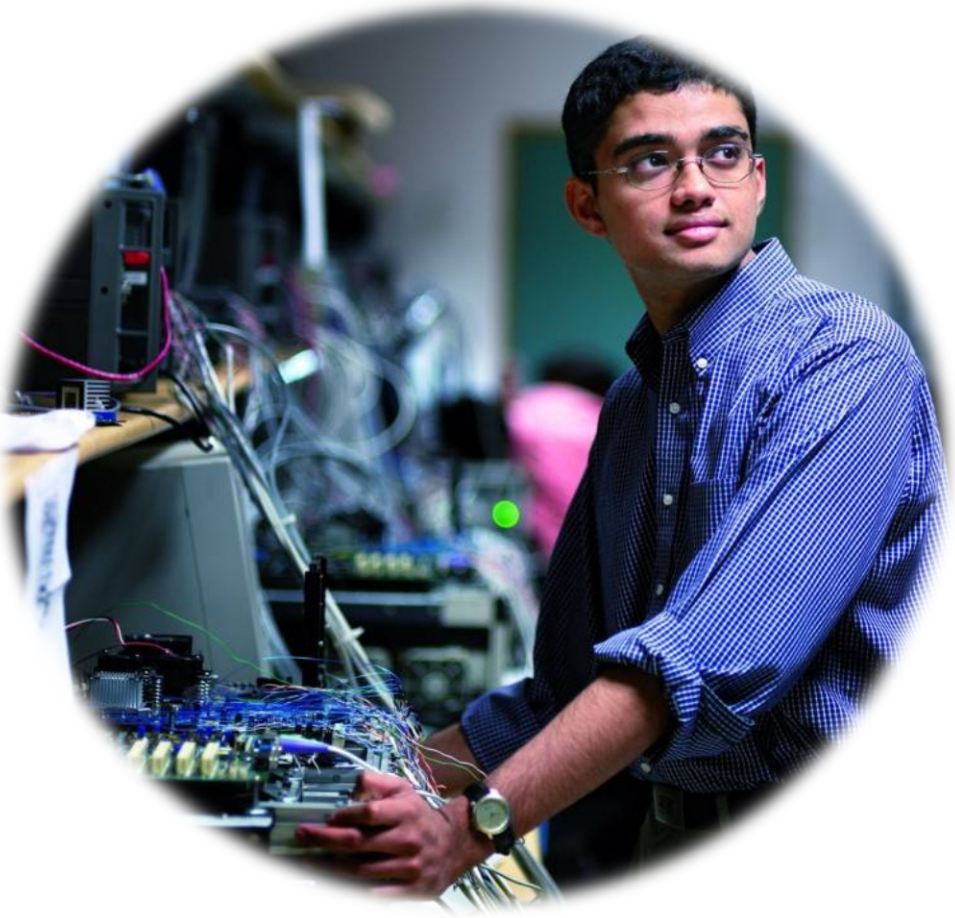
Firmware Debug Tool Wishlist

Common ground between developers & field technicians

The Developer	The Field Technician
• Use standalone or with another PC	• Use standalone or with another PC
• Use w/o opening case	• Use w/o opening case
• View checkpoints	• View checkpoints
• Store data for analysis	• Store data for analysis
• Use on production HW	• Use on production HW
• View debug strings	• No proprietary ports
• Source-level debug	

New Platform Designs Demand New Debug Tools

Agenda



Limitations for UEFI Debugging

Utilizing USB Debug Solutions

Extending UEFI Debugging Concepts

Using USB Debugging in the Field

Utilizing USB Debug Solutions

Why USB?

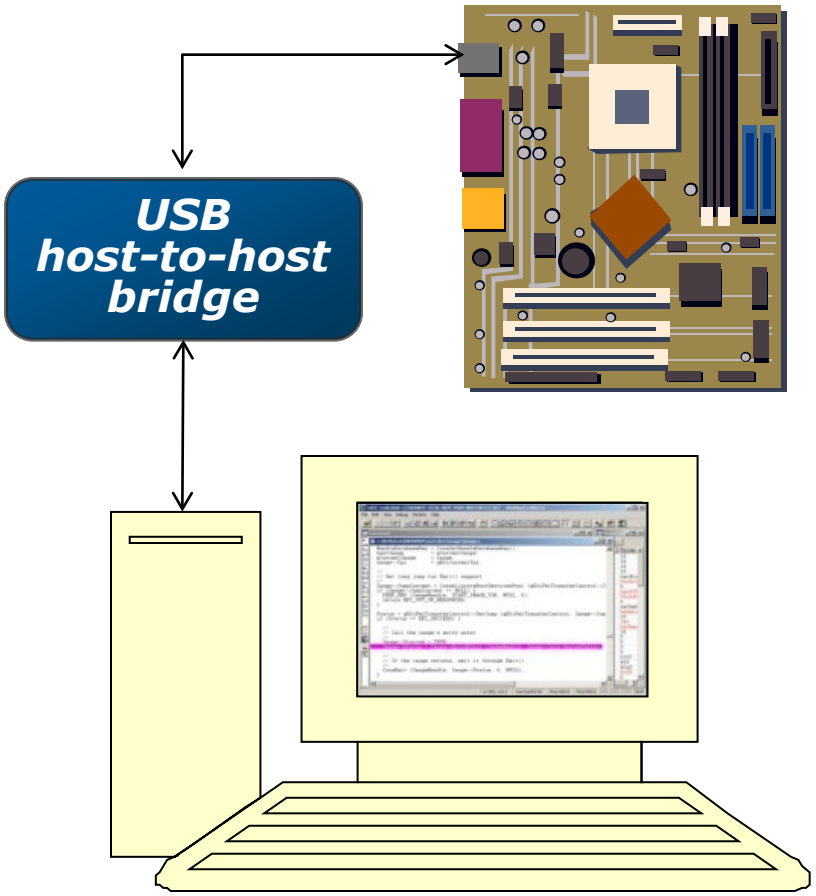
- USB is Ubiquitous
- Externally Accessible, Screwdriver Free
- USB 2.0 Enables Early Debugging via the EHCI debug port
- Same port works with debug devices or standard USB devices

What's a "debug port"

- One USB port supporting a simplified USB protocol
  - Fast protocol
  - Does not require full memory stack
  - Works only with "debug descriptor" device
- Supported by Intel ICH/SCH with USB 2.0

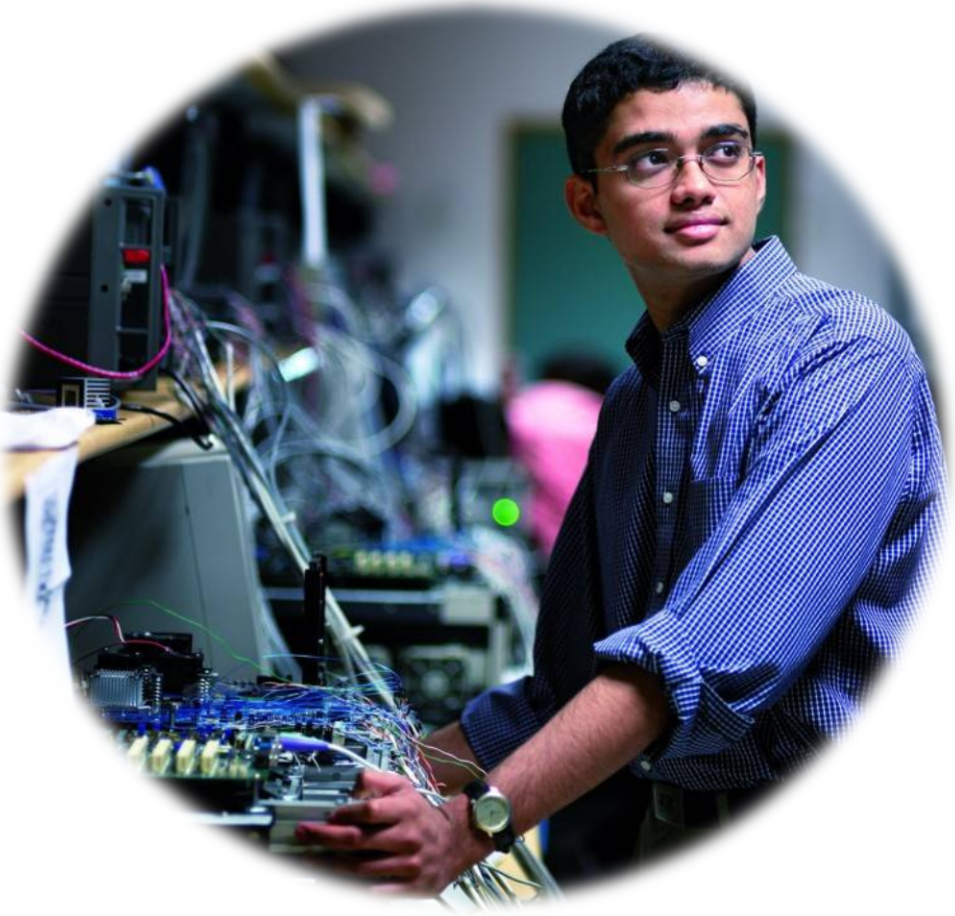
Today's Uses in Source Debugging

- USB Debug Port works as a "transport layer"
  - UEFI Debug Protocol
  - Requires host-to-host bridge
- Shown previously at IDF
- Example: AMI Debug
  - Source-level debug
  - DXE, PEI and UEFI Shell
  - Add breakpoints
  - Read & write mem/IO/PCI
  - Redirect debug messages
  - Redirect remote console



USB Debug Port Is Already Available & Used by IBVs

Agenda



Limitations for UEFI Debugging

Utilizing USB Debug Solutions

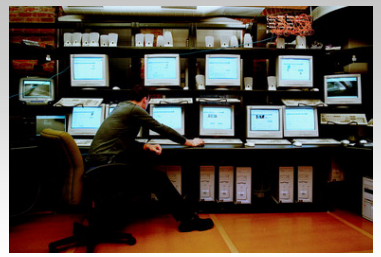
Extending UEFI Debugging Concepts

Using USB Debugging in the Field

Extending UEFI Debugging Concepts

Field Technicians

- Diagnose systems using checkpoints or status codes.
- Translate "hexadecimal nerd nonsense" into usable data.



Quality Assurance

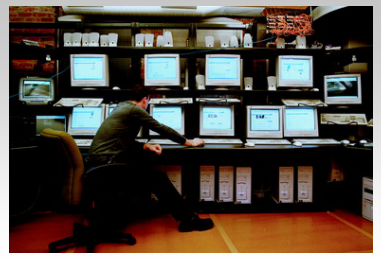
- Measure boot performance using checkpoint timing
- Record data for test reports
- Easily pinpoint hangs

BIOS/UEFI Developers

- Read checkpoints
- Optimize boot performance using checkpoint timing
- Enable source-level debug

Extending UEFI Debugging Concepts

Field Technicians



Quality Assurance

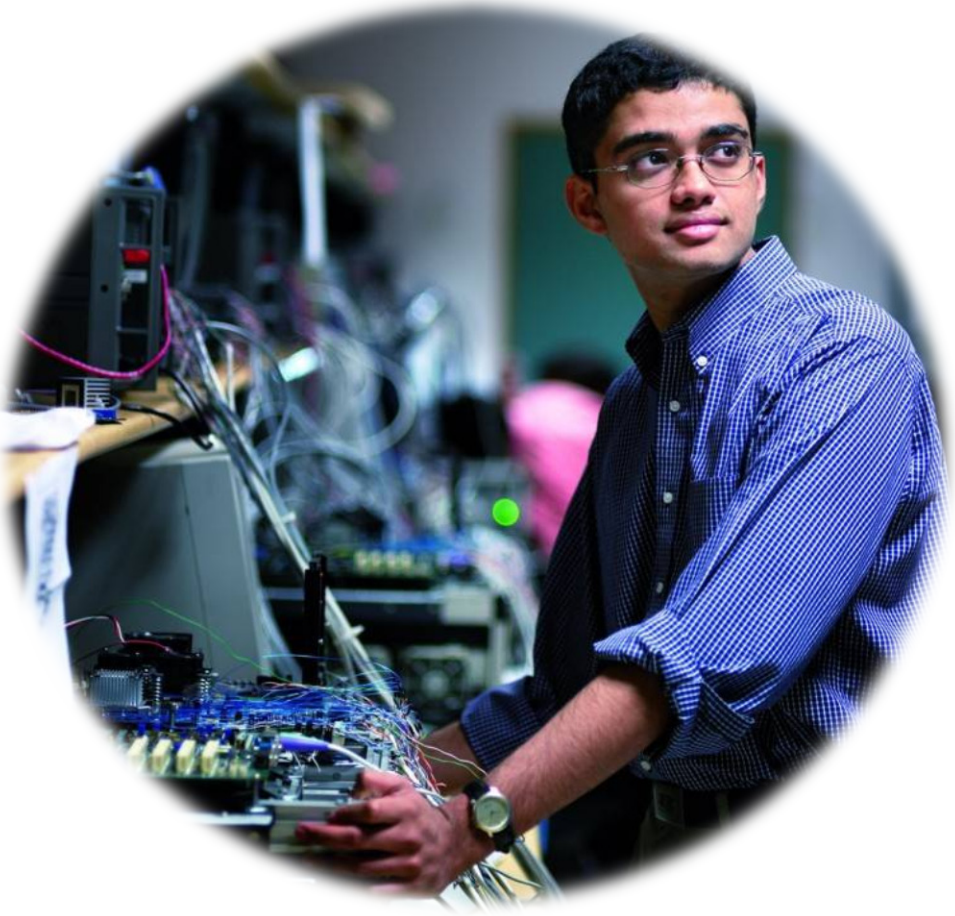
BIOS/UEFI Developers

For years, the focus has been on fixing problems for *BIOS developers*

There are *new product opportunities* solving the same set of problems for QA & field technicians

New Tools in UEFI Can Go Beyond Traditional BIOS Debugging

Agenda



Limitations for UEFI Debugging

Utilizing USB Debug Solutions

Extending UEFI Debugging Concepts

Using USB Debugging in the Field

Using USB Debugging in the Field

An Example Based on Today's Tools from AMI

- Stand-Alone Operation
  - Read & store checkpoints
  - Store UEFI debug strings
  - Replace cryptic hex values with text descriptions
  - Measure boot timing
- Use with Another PC
  - Stream UEFI debug strings live to a console
  - Enable source-level debug
  - Access stored sessions
  - Enabled in firmware by drop-in modules

AMIDebug™ Rx

Connects to USB EHCI Debug Port

Enhanced Features in USB Debug

- UEFI Debug Strings
  - Used when BIOS is compiled in "debug mode"
  - Pass strings in DEBUG() & ASSERT() macros
  - Better information than just checkpoints
  - Redirected to AMI Debug Rx & USB Debug Port

```
[AmiDbg]Register PPI Notify: f894643d-c449-42d1-8ea8-85bdd8c65bde
[AmiDbg]Register PPI Notify: 605ea650-c65c-42e1-ba80-91a52ab618c6
[AmiDbg]CpuPeiBeforeMem.Entry(FFFECB85)
[AmiDbg]NBPEI.Entry(FFFF495B)
[AmiDbg]SBPEI.Entry(FFFF1AED)
[AmiDbg]>>> PM Registers Before GPIO Init <<<
[AmiDbg]+===================== PM Registers dump ======================+
[AmiDbg]   PM1a_EVT_BLK.PM1_STS       : Addr = 0400 => Val = 0001
[AmiDbg]   PM1a_EVT_BLK.PM1_EN        : Addr = 0402 => Val = 0000
```

Enhanced Features in USB Debug

- Boot Time Analysis
  - Used on any BIOS with AMI Debug Rx support
  - Based on device's internal timer
  - Total boot time or time between checkpoints

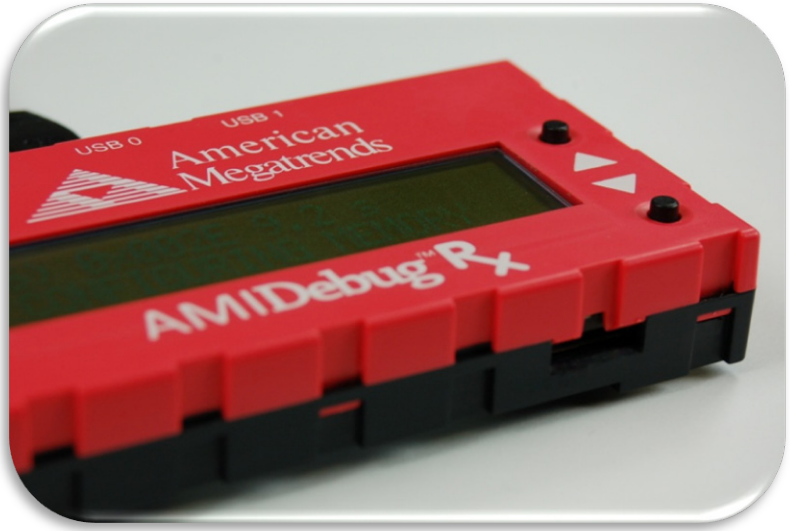
```
Session Start Time      :          06/10/2009 15:16:44
Total Checkpoints       :          52
Duration of last boot   :          23,703ms
BIOS Tag                :          0ABFL032
BIOS Type               :          Aptio 4.x
BIOS Build Time         :          05/11/2009 17:00:07

                          Checkpoint Output
-------------------------------------------------------------------------------
 Num          CP            Time (ms)                    String
-----       ------       --------------          --------------------
    1       0x0011              1,372ms           PRE-MEM CPU INIT
    2       0x0015              1,513ms           PRE-MEM NB INIT
    3       0x0019              1,883ms           PRE-MEM SB INIT
    4       0x002B              8,674ms           MEM INIT. SPD READ
```

Demo

AMI Debug Rx in use …

- Capture Checkpoints
- Retrieve Stored Checkpoint Session
- Boot Time Analysis
- Store UEFI Debug Strings



Problems Solved w/AMI Debug Rx

Works with any System Form Factor

- No PCI slot or LPC header
- Externally accessible
- Uses commodity USB port
- Utilizes existing technology in today's USB 2.0 EHCI controllers

Single Solution for Multiple Applications

- Standalone or with another PC
- Field Debug & Quality Assurance
- Measure boot performance
- Enable source-level debugging

IBV Debug Tools Can Support Products From Development to Deployment

Key Learnings

- New Platform Designs Demand New Debug Tools
- USB Debug Port Is Already Available & Used by IBVs
- New Tools in UEFI Can Go Beyond Traditional BIOS Debugging
- IBV Debug Tools Can Support Products From Development to Deployment

Next Steps –
Best Technical Methods for UEFI Development

- **UEFI is a rich environment visit the UEFI web site**
  - **Learning center on UEFI web site**
- **Down load the white papers**
- **Work with your IBVs for the latest innovation tools**

Additional resources on UEFI:

- Demos in the Showcase
  - UEFI Booth #136
  - American Megatrends Inc #429
- Talk to other UEFI members in the showcase
- Other information on the web
  - Boot Optimization Whitepaper: http://edc.intel.com/Link.aspx?id=2355
  - "Improving BIOS Debugging Using USB 2.0 Methods" Whitepaper available at www.ami.com
  - AMI Debug Rx product information at www.ami.com/amidebugrx
  - "USB2 Debug Device Functional Specification, Revision 0.90" available at www.intel.com
  - Specifications and Implementation sites: www.tianocore.org, www.uefi.org, www.intel.com/technology/efi
- Technical book from Intel Press:
  - "Beyond BIOS: Implementing the Unified Extensible Firmware Interface with Intel's Framework" www.intel.com/intelpress

IDF 2009 UEFI Sessions

EFI#	Company	Description	Time	RM	D
P001 ✓	Dell, HP, IBM, Intel, Microsoft	Using UEFI as the Foundation for Innovation	10:15	2005	T
S001 ✓	IBM, Intel	Intel Advanced Technology in the Enterprise: Best Security Practices	16:15	2001	W
S002 ✓	Dell, Intel, Insyde SW	Secure FW Lockdown through Standardized UEFI Management Protocols	17:15	2001	W
S003 ✓	Intel, AMI	Best Technical Methods for UEFI Development -Reducing Platform Boot Times -Firmware Debugging: UEFI and USB for platform forensics	11:10	2002	Th
S004	Microsoft, Insyde SW, Intel	UEFI Boot Time Opt. Under Microsoft Windows 7	13:40	2002	Th
S005	Phoenix, Intel	Transitioning the Plug-In Industry from Legacy to UEFI: Real World Cases	14:40	2002	Th
Q001	Intel, All	UEFI Q & A session	15:40	2002	Th

✓ DONE

Session Presentations - PDFs

The PDF for this Session presentation is available from our IDF Content Catalog at the end of the day at:

intel.com/go/idfsessions

Please Fill out the Session Evaluation Form

Give the completed form to the room monitors as you exit!

Thank You for your input, we use it to improve future Intel Developer Forum events

Q&A

Legal Disclaimer

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.
- Intel may make changes to specifications and product descriptions at any time, without notice.
- All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
- Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.
- Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.
- *Other names and brands may be claimed as the property of others.
-

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the third quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the corporation's expectations. Ongoing uncertainty in global economic conditions pose a risk to the overall economy as consumers and businesses may defer purchases in response to tighter credit and negative financial news, which could negatively affect product demand and other related matters. Consequently, demand could be different from Intel's expectations due to factors including changes in business and economic conditions, including conditions in the credit market that could affect consumer confidence; customer acceptance of Intel's and competitors' products; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Additionally, Intel is in the process of transitioning to its next generation of products on 32nm process technology, and there could be execution issues associated with these changes, including product defects and errata along with lower than anticipated manufacturing yields. Revenue and the gross margin percentage are affected by the timing of new Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel's response to such actions; and Intel's ability to respond quickly to technological developments and to incorporate new features into its products. The gross margin percentage could vary significantly from expectations based on changes in revenue levels; capacity utilization; start-up costs, including costs associated with the new 32nm process technology; variations in inventory valuation, including variations related to the timing of qualifying products for sale; excess or obsolete inventory; product mix and pricing; manufacturing yields; changes in unit costs; impairments of long-lived assets, including manufacturing, assembly/test and intangible assets; and the timing and execution of the manufacturing ramp and associated costs. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. The current financial stress affecting the banking system and financial markets and the going concern threats to investment banks and other financial institutions have resulted in a tightening in the credit markets, a reduced level of liquidity in many financial markets, and heightened volatility in fixed income, credit and equity markets. There could be a number of follow-on effects from the credit crisis on Intel's business, including insolvency of key suppliers resulting in product delays; inability of customers to obtain credit to finance purchases of our products and/or customer insolvencies; counterparty failures negatively impacting our treasury operations; increased expense or inability to obtain short-term financing of Intel's operations from the issuance of commercial paper; and increased impairments from the inability of investee companies to obtain financing. The majority of our non-marketable equity investment portfolio balance is concentrated in companies in the flash memory market segment, and declines in this market segment or changes in management's plans with respect to our investments in this market segment could result in significant impairment charges, impacting restructuring charges as well as gains/losses on equity investments and interest and other. Intel's results could be impacted by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. A detailed discussion of these and other risk factors that could affect Intel's results is included in Intel's SEC filings, including the report on Form 10-Q for the quarter ended June 27, 2009.

Rev. 7/27/09