The IA-64 System Architecture: Tutorial For Hardware, OS, & Application Developers

February 15, 2000

Jerry Huck
Manager of Processor Architecture
Computer Products Group
Hewlett Packard Company

Rumi Zahir
Itanium™ Processor System Architecture
Microprocessor Products Group
Intel Corporation

What's In This Session?

- IA-64 memory models, system software, other architecture features relevant to hardware, OS, and application developers
- IA-64 concepts & features for developers to take advantage and build robust & scalable software and hardware components

IA-64 System Architecture Agenda

- System Architecture Highlights
- Virtual Memory Model
- Interruption Model
- System Software Stack
- Reliability, Availability, Serviceability
- Parallelism & Scalability
- Compatibility
- Summary

IA-64 System Architecture Goals

- Flexible Architecture
  - "address space per process" model
    - Windows NT, Unix, Mach, etc.
  - "Global Address Space" model
    - HP-UX plus future 64-bit OSes
- Performance focused Hardware
- Support shrink-wrap OS compatibility
- Scalable system performance
- IA-32 and PA-RISC Compatibility

IA-64 System Architecture Agenda

- System Architecture Highlights
- Virtual Memory Model
- Interruption Model
- System Software Stack
- Reliability, Availability, Serviceability
- Parallelism & Scalability
- Compatibility
- Summary / Call To Action

IA-64 Virtual Memory Model

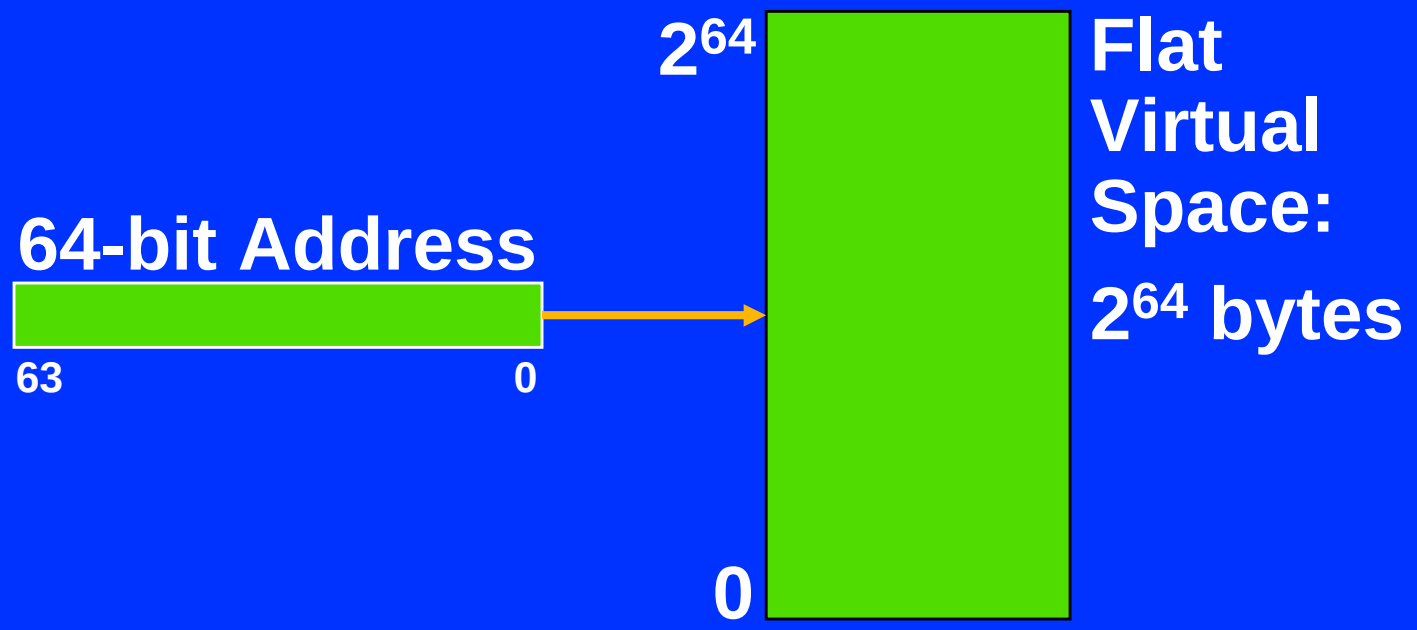
- Process Address Space
- System Address Space Management
- Virtual Address Translation
  - TLB and Page table
- Flexible Object Sharing Model
  - Aliasing and Global addressing

64-bit Address Space: Large and Sparse

- Requires increased TLB capacities
  - Mapped files, globally shared memory
  - Allow for multiple large on/off-chip TLBs
  - Provide wide range of page sizes
- Improve utilization and sharing of TLB and page tables
  - No TLB flush on context switch
  - Promote TLB entry sharing

IA-64 Overcomes Challenges Of 64-bit Address Space Efficiently

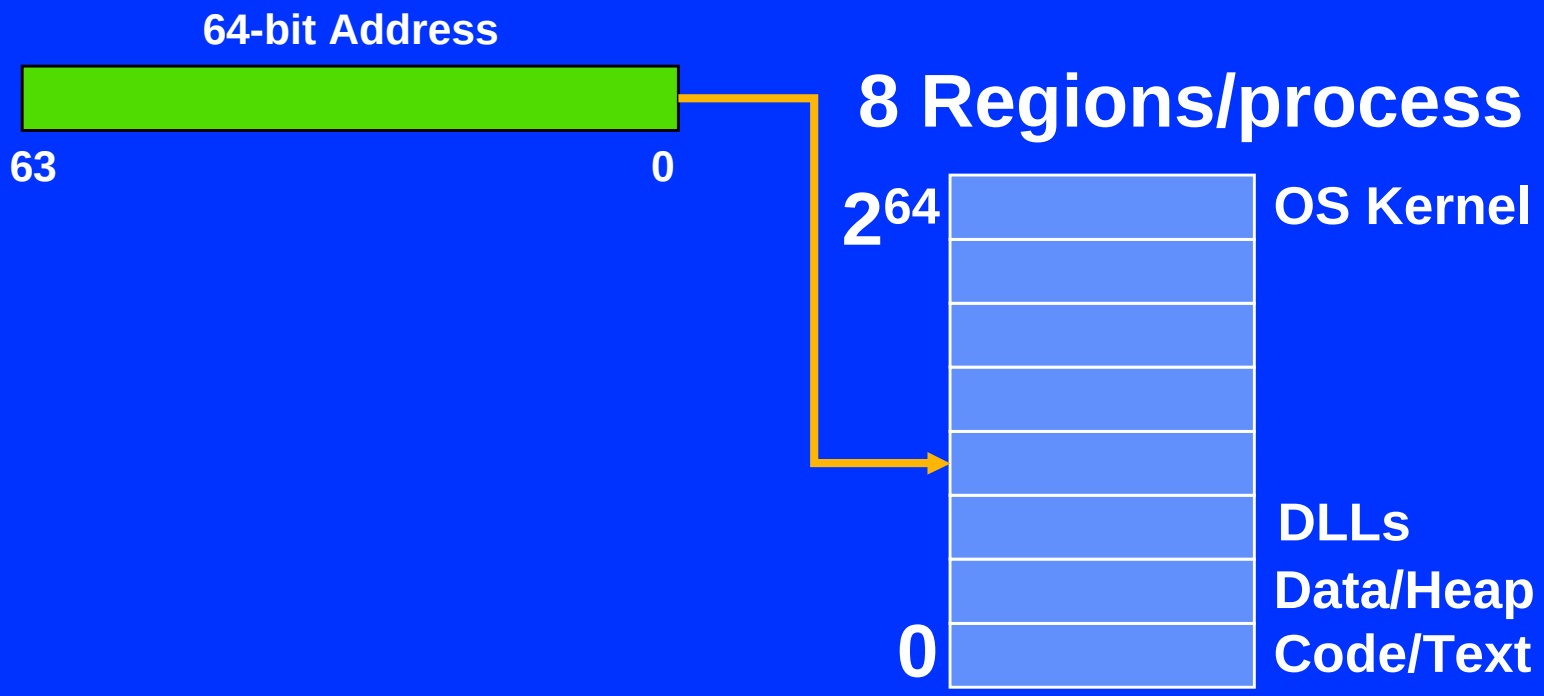
Process Address Space



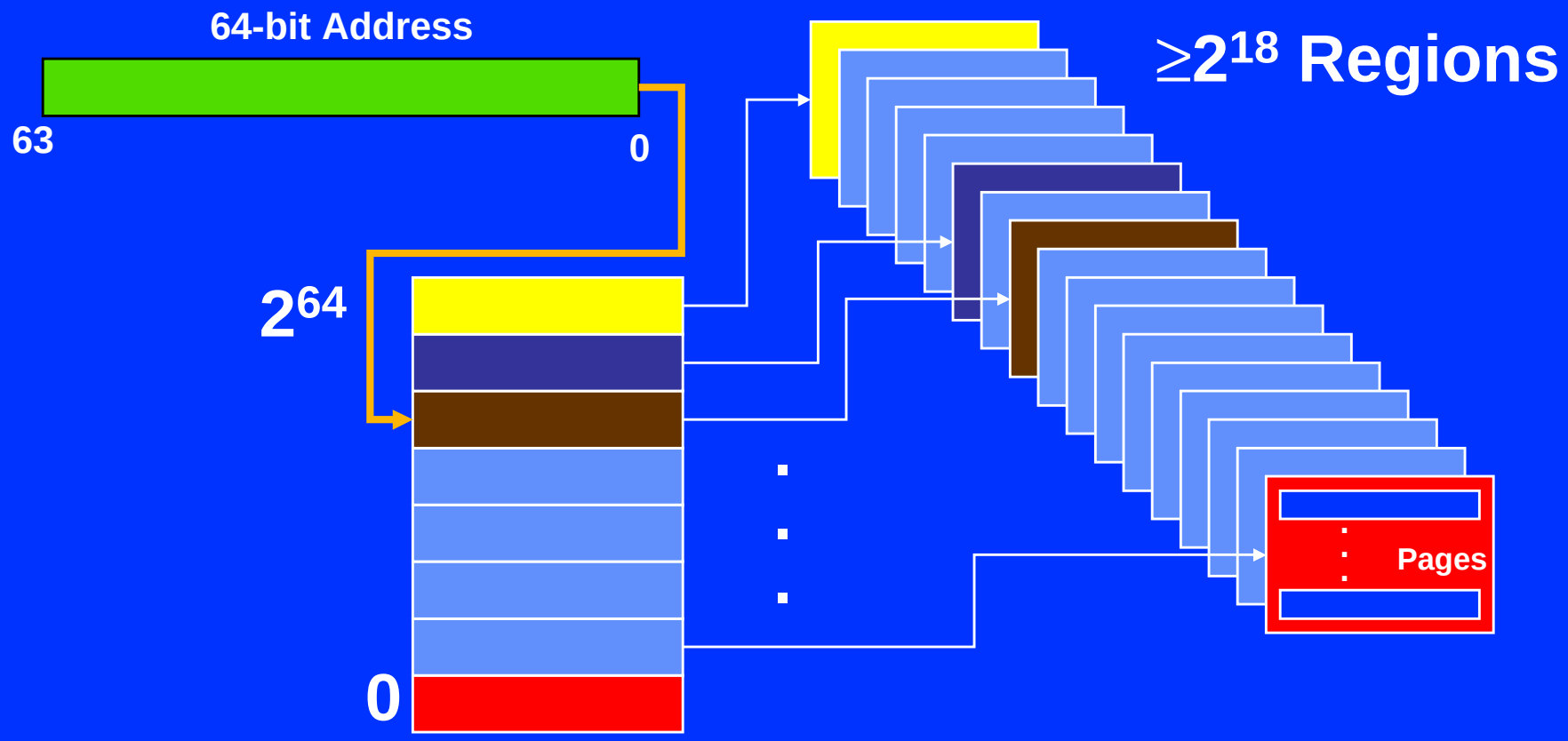
invent

intel

Process Address Space



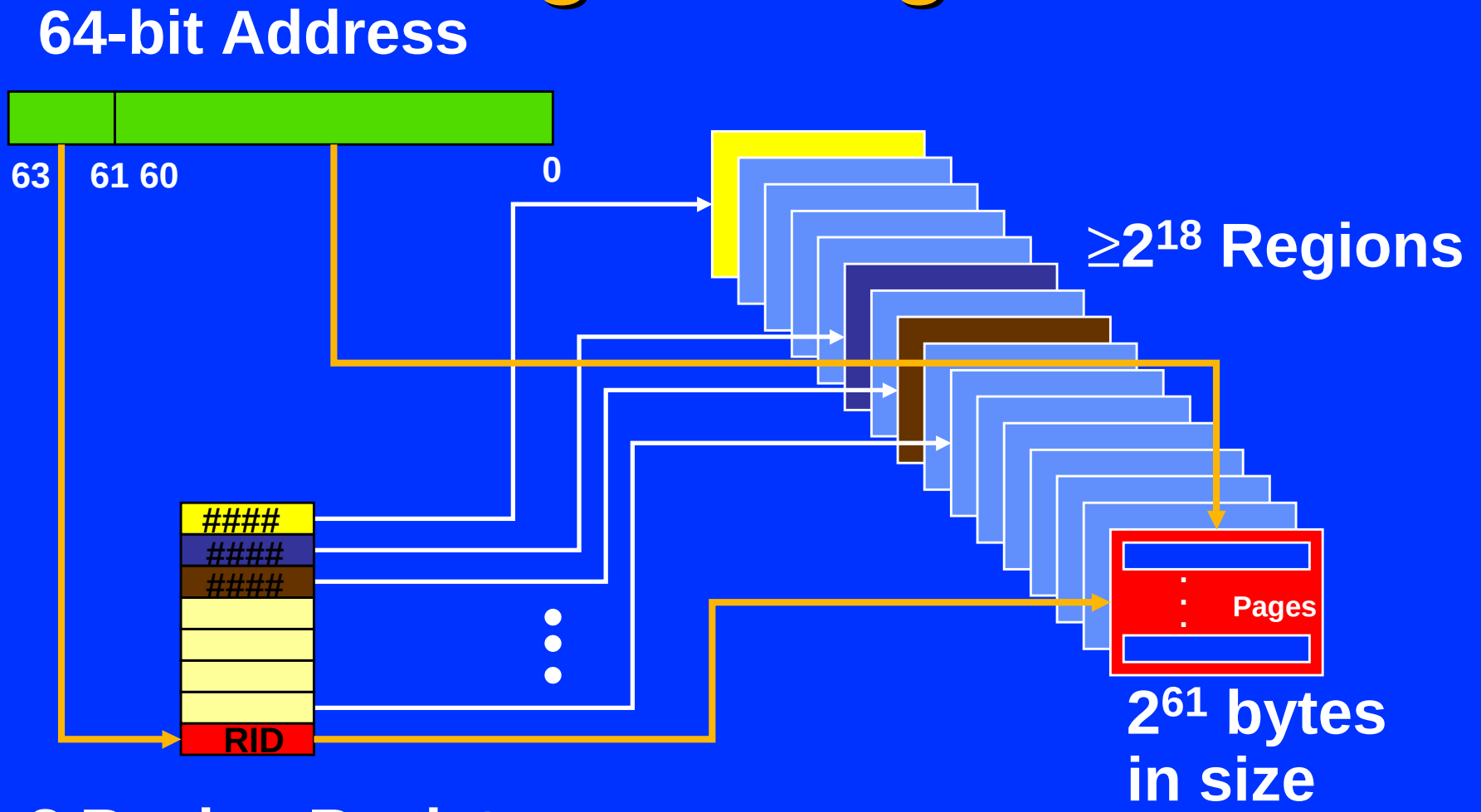
System Address Space



invent

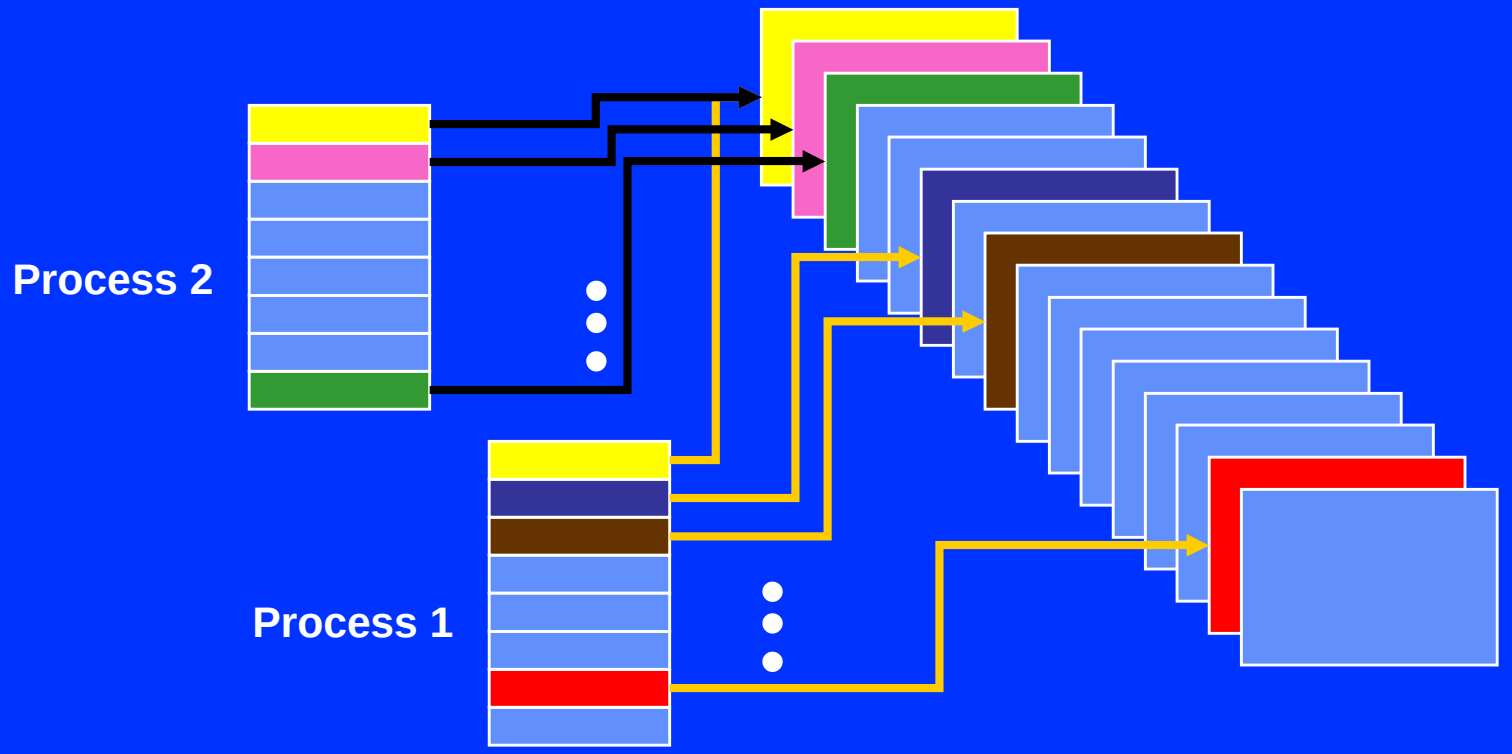
intel

IA-64 Region Registers



invent

Processes and Threads



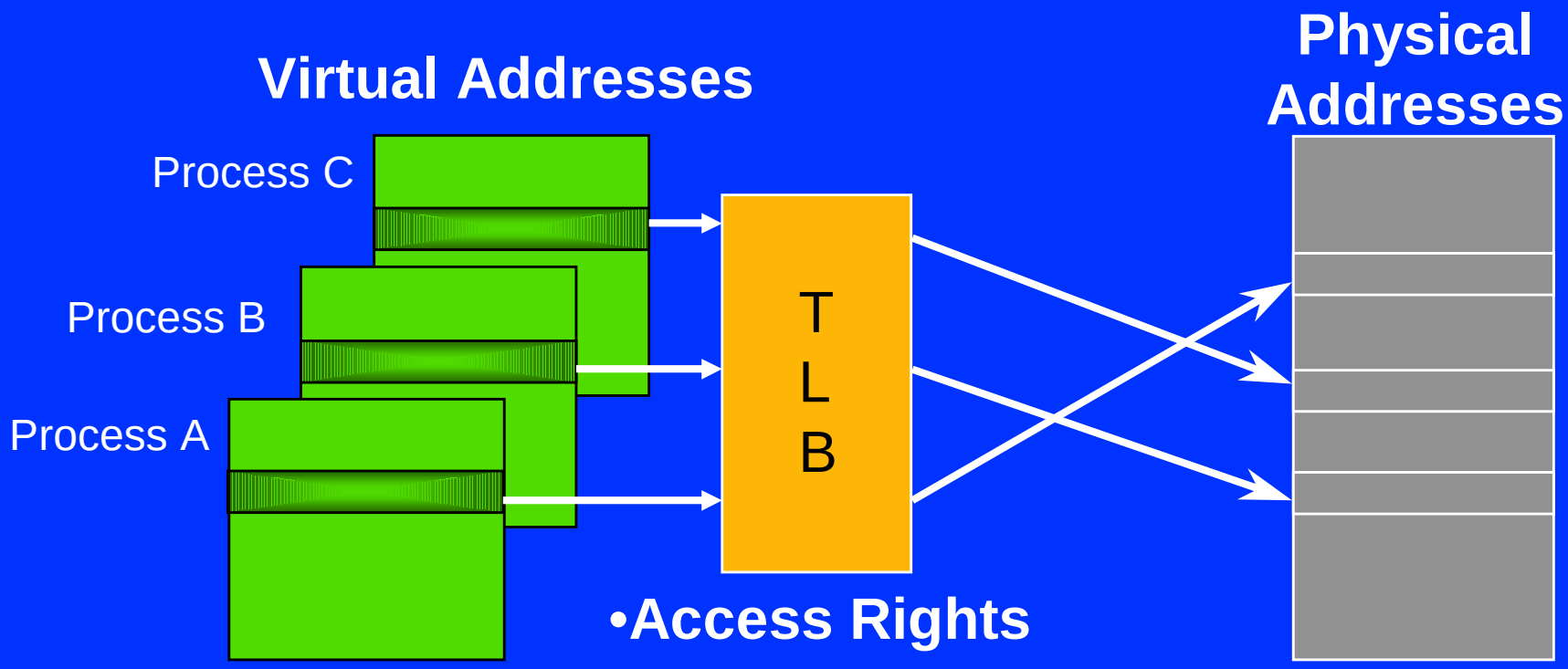
Regions Enable Efficient Management Of Processes For Multi-tasking Environments

invent

intel

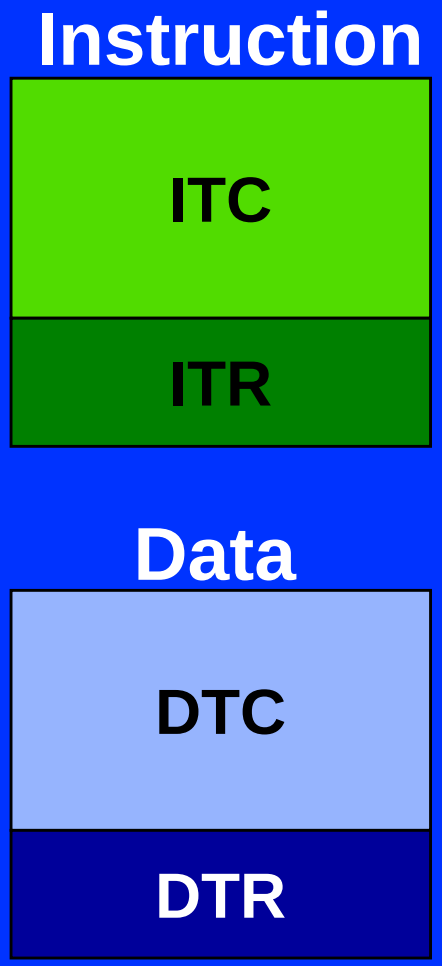
Virtual Address Translation: TLB

- Mapping to Physical Address



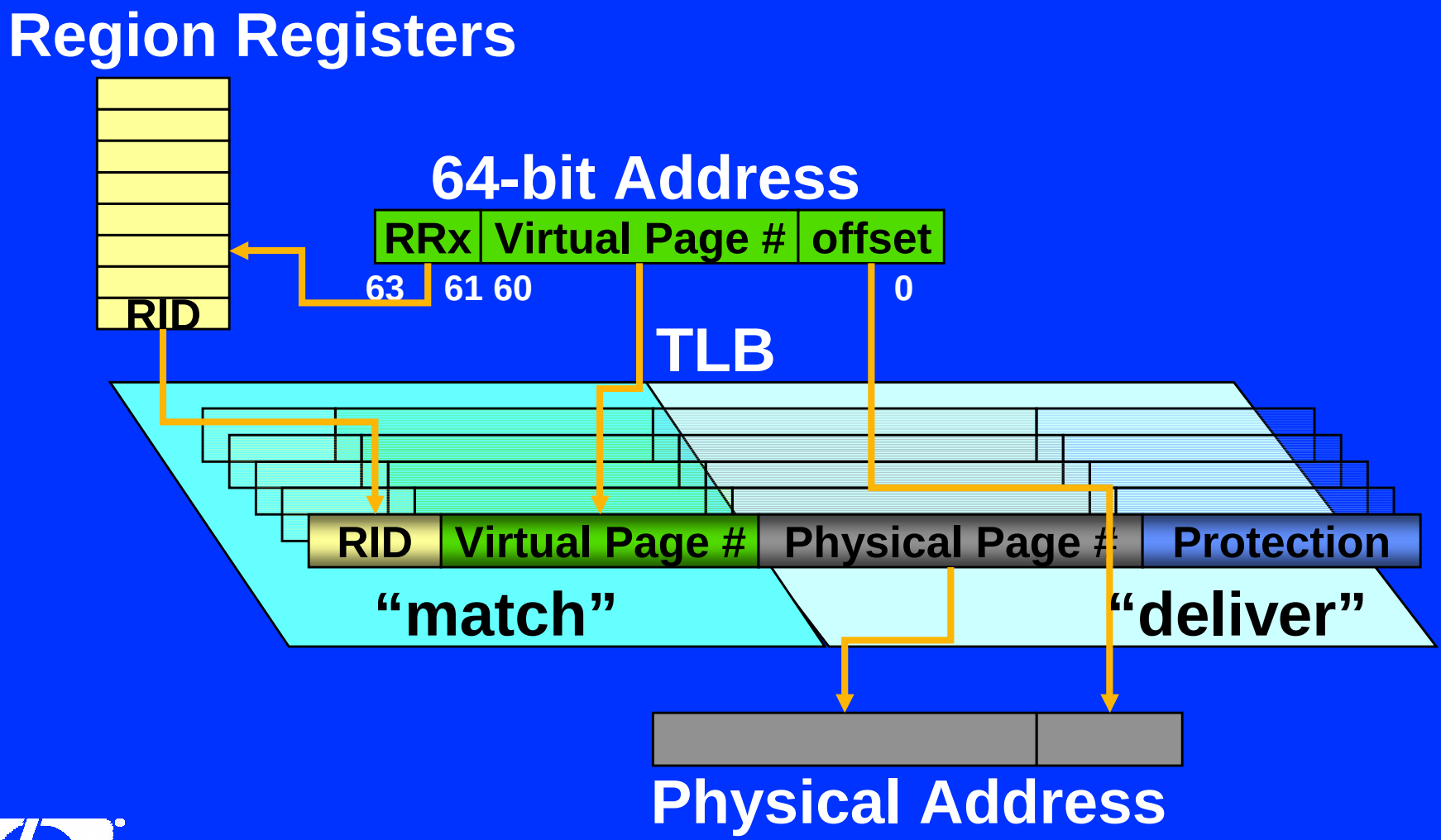
TLB Organization

- Separate instruction and data TLBs
- Software Manages
  - TR entries,
  - Page-table updates
- Hardware Manages
  - TC TLB refill
  - Broadcast TLB Purge

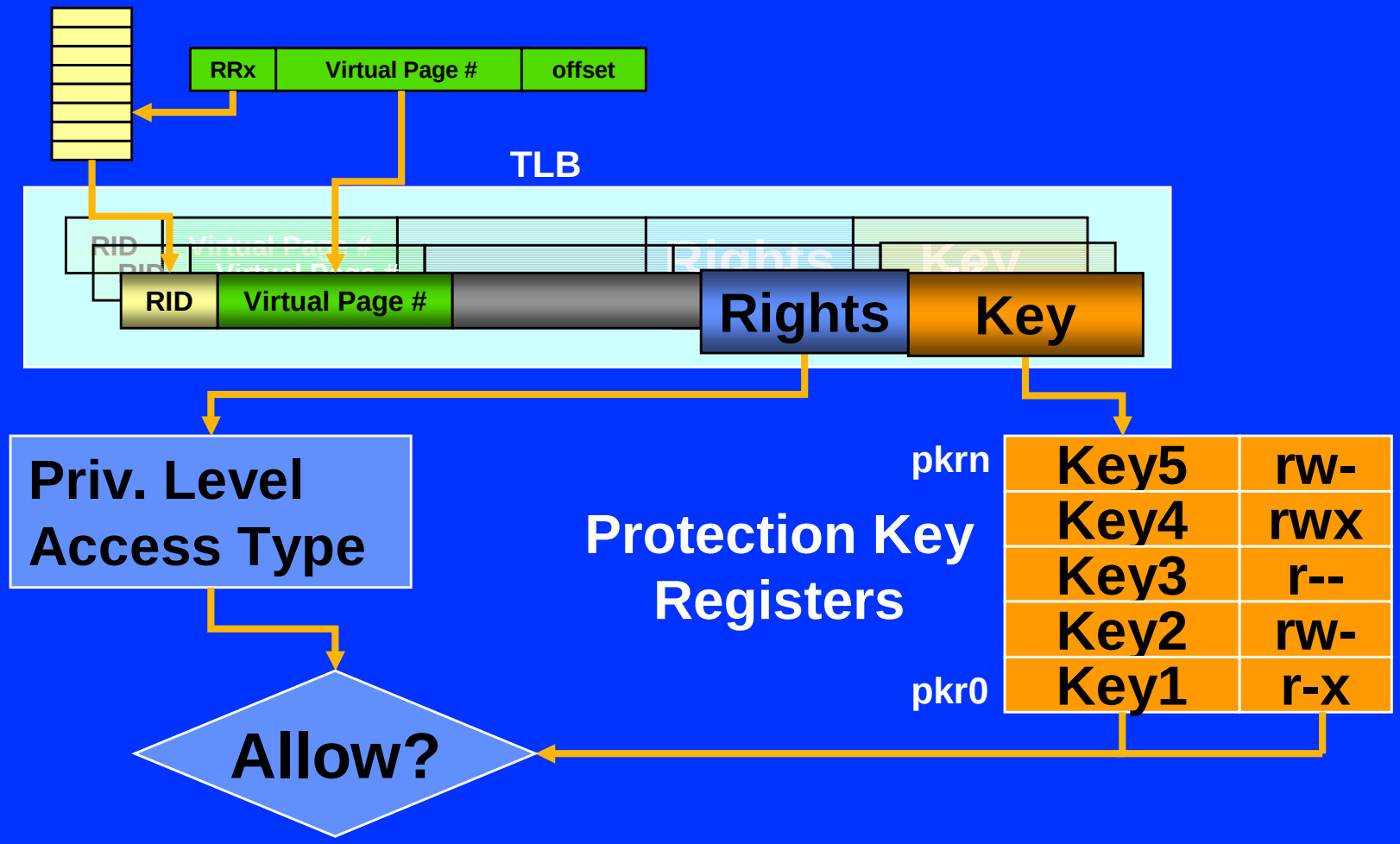


Balance TLB For Efficient Memory Management

Virtual Address Translation



Protection: Can I See it? Can I Access it?



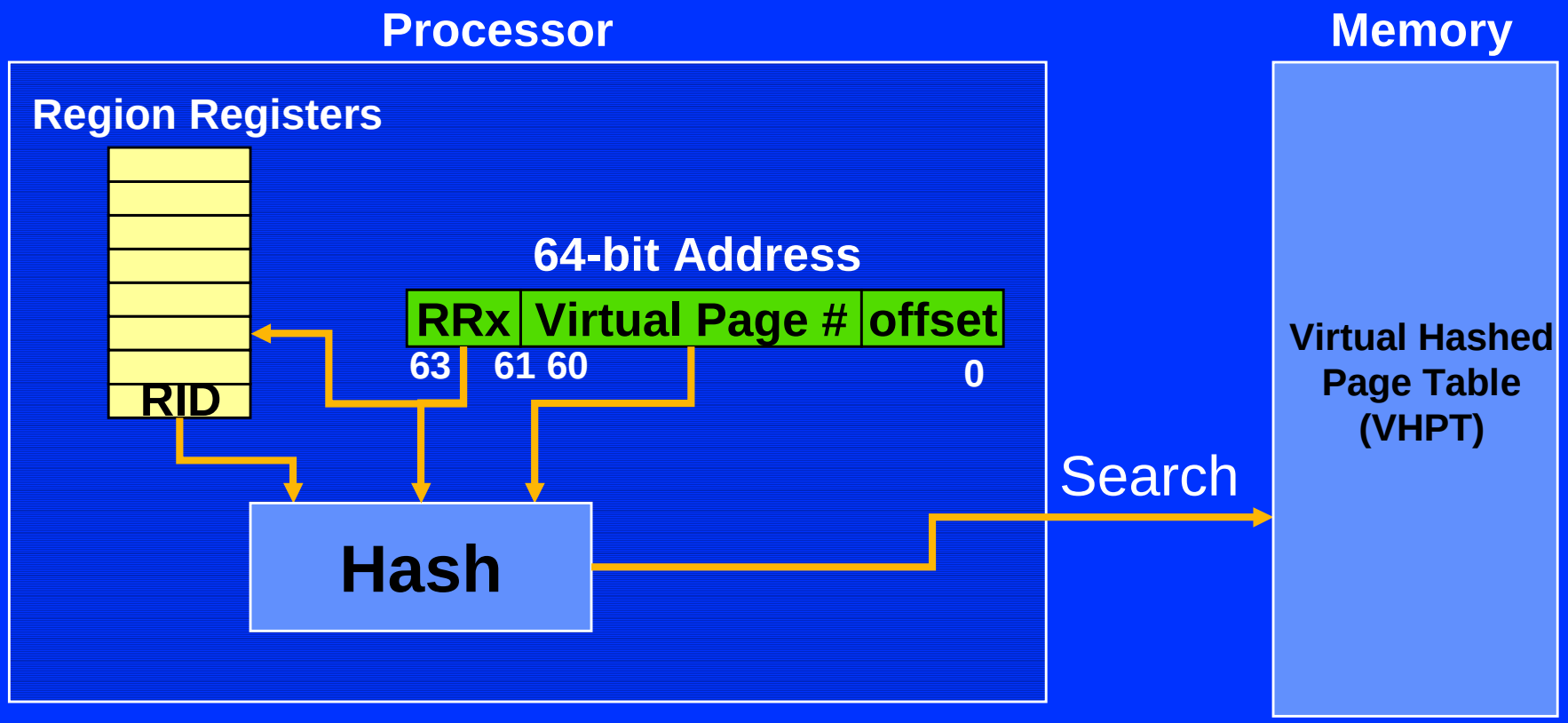
Protection Keys Increase TLB Utilization For Large Object Databases

Variable Page Sizes

- Minimum on all implementations
  - 4K, 8K, 16K, 64K, 256K, 1M, 4M, 16M, 64M, 256M-bytes
- 4 GB purge
  - Simplify address space de-allocation

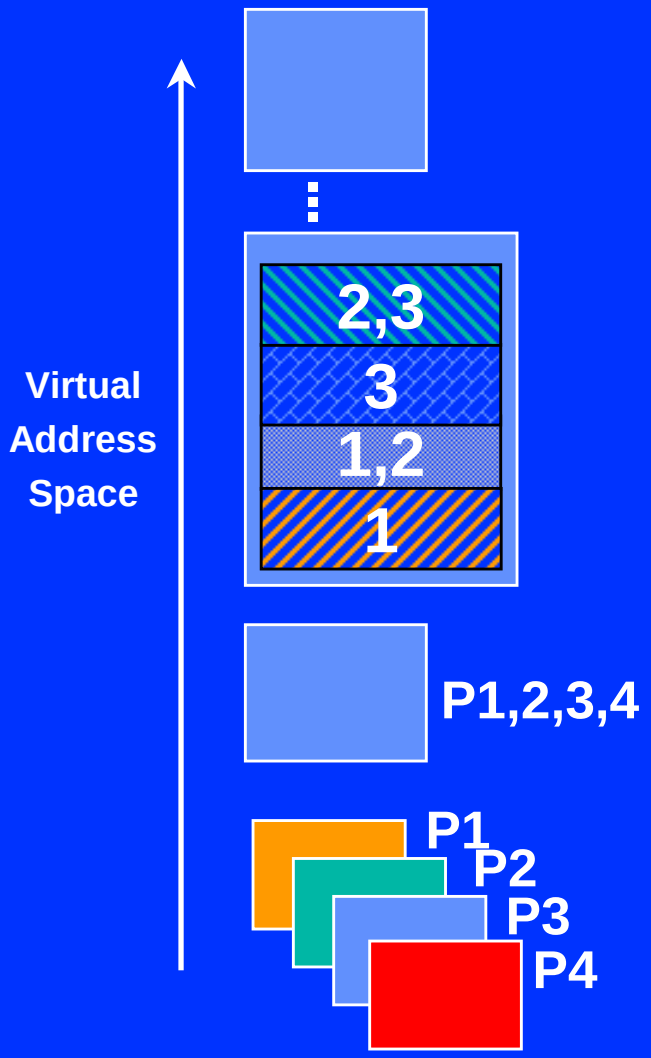
Variable Page Sizes Enable TLB Efficiency For OS And Application Performance

Hardware Accessed Page Table



Flexible Hardware Mechanisms Enable Parallel Execution

Virtual Memory Model: Example



Region 7 - One RID, no key

Kernel - protected by Priv. level

Region 2 - One RID, protection via multiple keys

Shared memory areas

Region 1 - Same RID if shared

Single address space for code

Region 0 - Different RID in each process

Unique address spaces for data

Flexible Virtual Memory Architecture Enables Variety Of Efficient OS Implementations

hp invent

intel

IA-64 System Architecture Agenda

- System Architecture Highlights
- Virtual Memory Model
- **Interruption Model**
- System Software Stack
- Reliability, Availability, Serviceability
- Parallelism & Scalability
- Compatibility
- Summary / Call To Action

IA-64 Interruption Model

- **Parallel instruction execution, . . .**
  - **Exception delivery is sequential & precise**
    - **All exceptions reported on the excepting instruction (including numeric exceptions)**
- **“Interruption” is IA-64 term for...**

Abort	Fault	Trap	Interrupt
• Hardware reset • Machine check	Exception taken before instruction commit, e.g. TLB miss	Exception taken after instruction commit, e.g. FP trap	Asynchronous external event: • device or platform management interrupt • soft-reset

IA-64 Provides Precise Exception Model To Match Today’s OS Designs

IA-64 Interruption Process

Normal Instruction Execution Flow:
- *Instruction A executed*

Application Code

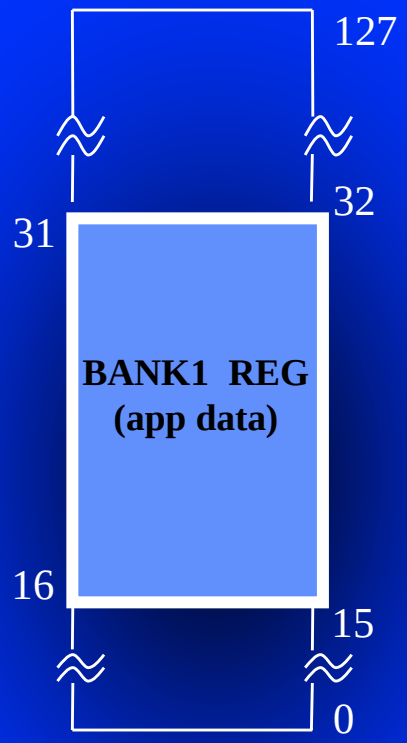
IP →	0x1000	INST A
	0x1010	INST B
	0x1020	INST C
		.
		.
		.

IVT Code

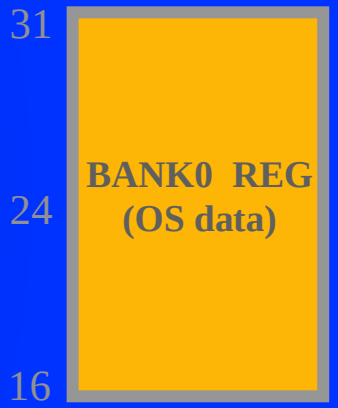
0x4000	INST X
0x4010	INST Y
0x4020	RFI
	.
	.

Current Processor State

IP	0x1000
PSR	



Interruption Registers



invent

intel

IA-64 Interruption Process

Normal Instruction Execution Flow:
- *Instruction B executed*

Application Code

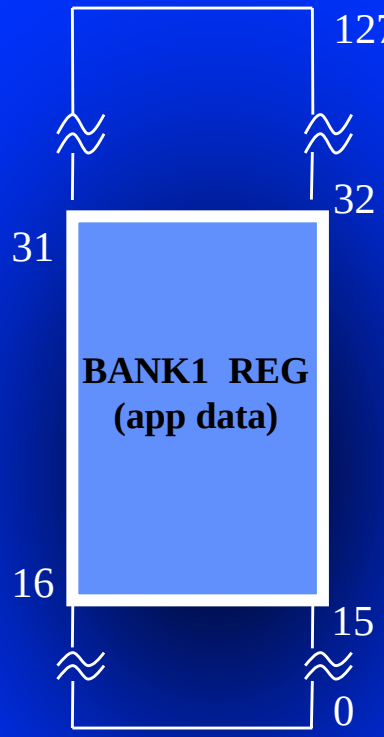
	0x1000	INST A
IP →	0x1010	INST B
	0x1020	INST C
		.
		.
		.

IVT Code

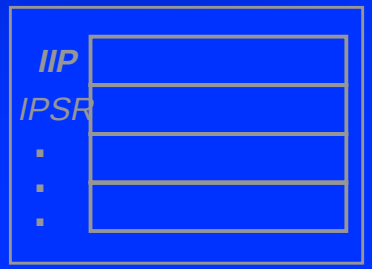
	0x4000	INST X
	0x4010	INST Y
	0x4020	RFI
		.
		.

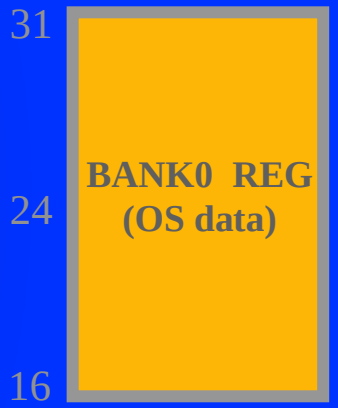
Current Processor State

IP	0x1010
PSR	



Interruption Registers



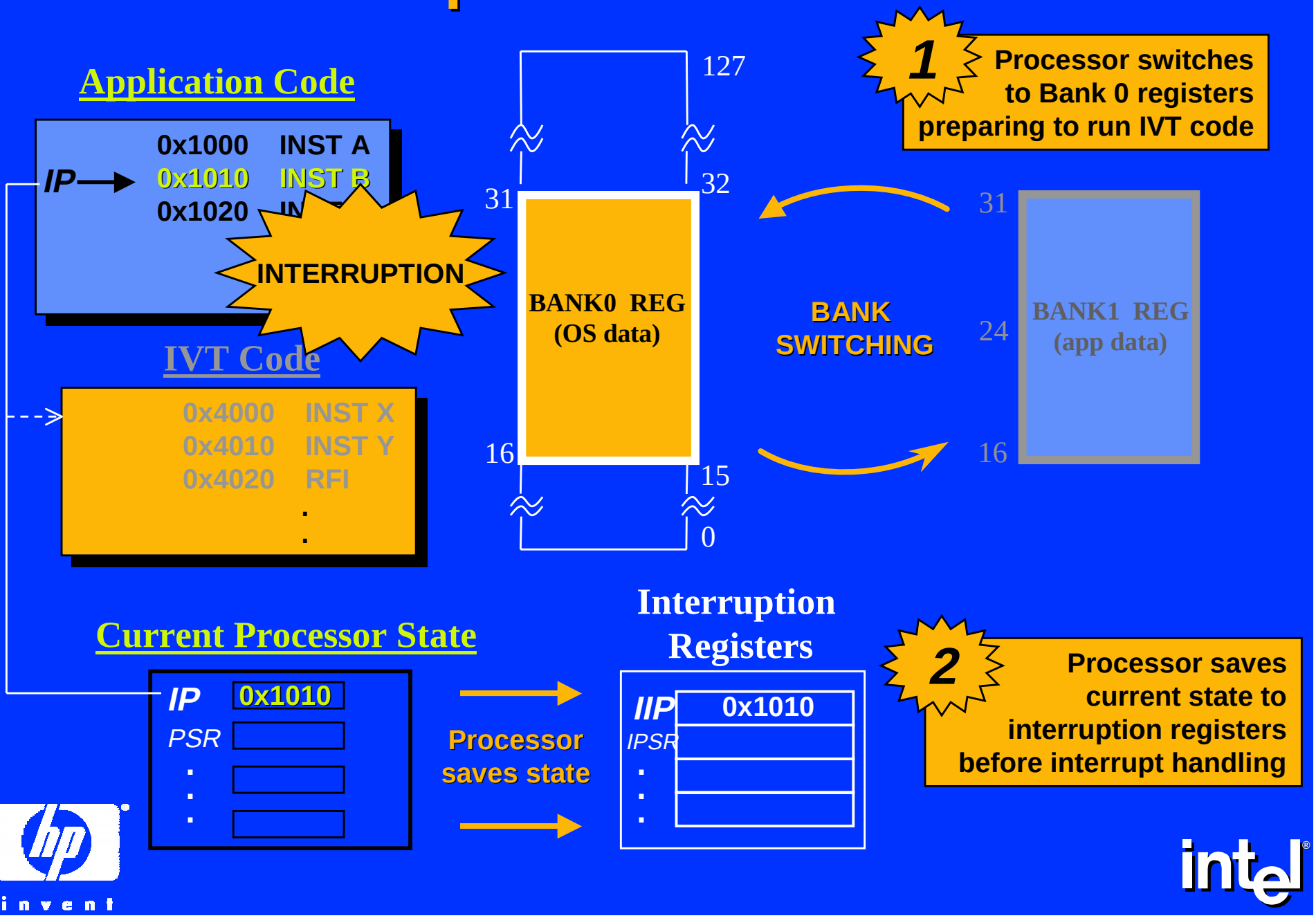


hp invent

intel

IA-64 Interruption Process

Interruption Delivery



invent

intel

IA-64 Interruption Process

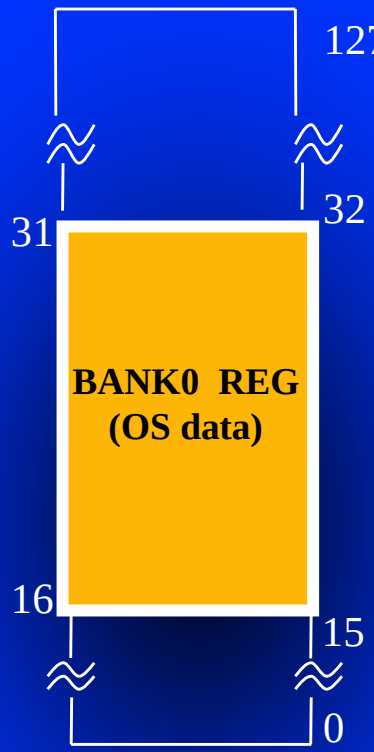
Interruption Handling
- *Instruction X executed in interrupt vector table*

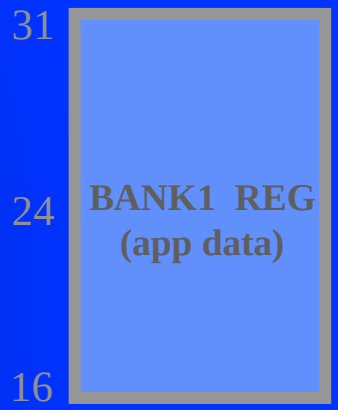
Application Code

0x1000 INST A
0x1010 INST B
0x1020 INST C
.
.

Interrupt Vector Table (IVT) Code

IP → 0x4000 INST X
0x4010 INST Y
0x4020 RFI
.
.





Current Processor State

IP 0x4000
PSR
.
.
.

Interruption Registers

IIP 0x1010
IPSR
.
.
.

invent

intel

IA-64 Interruption Process

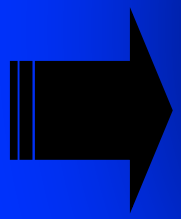
Interruption Handling
- *Instruction Y executed in interrupt vector table*

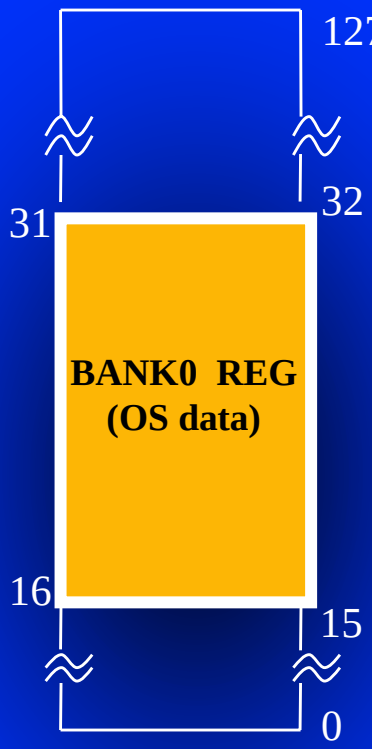
Application Code

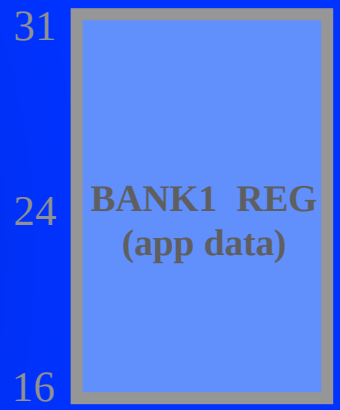
0x1000 INST A
0x1010 INST B
0x1020 INST C
.
.

IVT Code

IP → 0x4000 INST X
0x4010 INST Y
0x4020 RFI
.
.







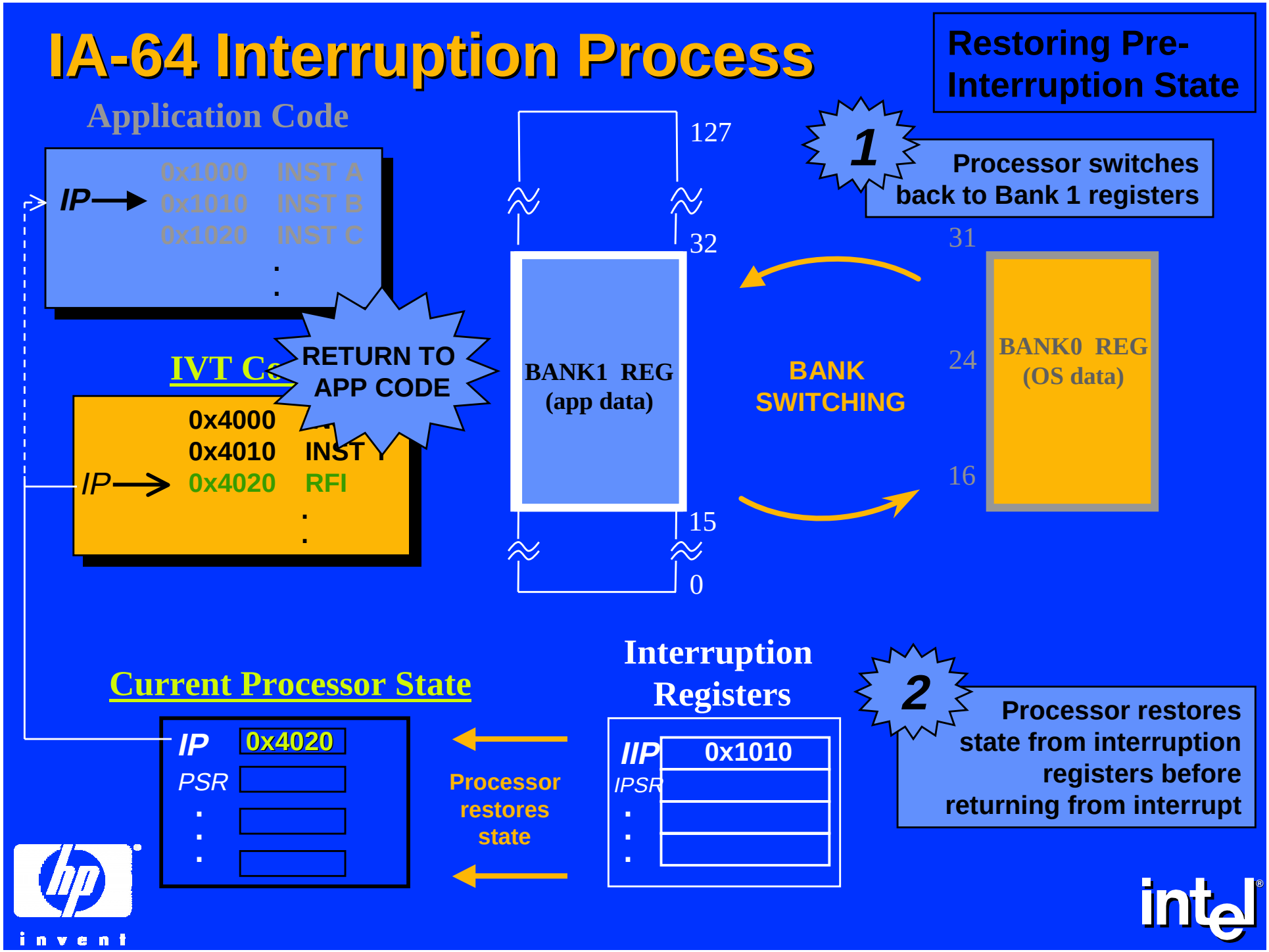
Current Processor State

IP 0x4010
PSR
.
.
.

Interruption Registers

IIP 0x1010
IPSR
.
.
.

IA-64 Interruption Process



IA-64 Interruption Process

Resume Normal Instruction Execution:
- *Instruction B executed*

Application Code

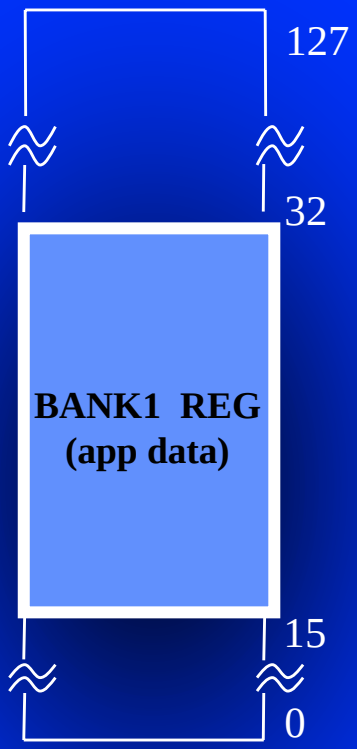
	0x1000	INST A
IP →	0x1010	INST B
	0x1020	INST C
		.
		.
		.

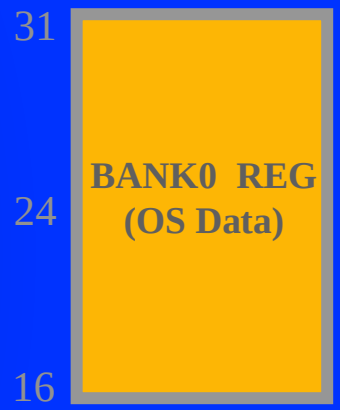
IVT Code

	0x4000	INST X
	0x4010	INST Y
	0x4020	RFI
		.
		.

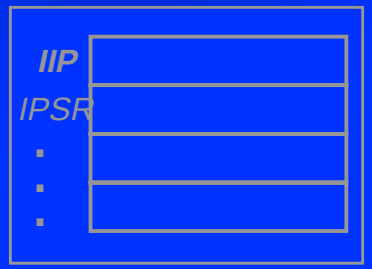
Current Processor State

IP	0x1010
PSR	





Interruption Registers



hp invent

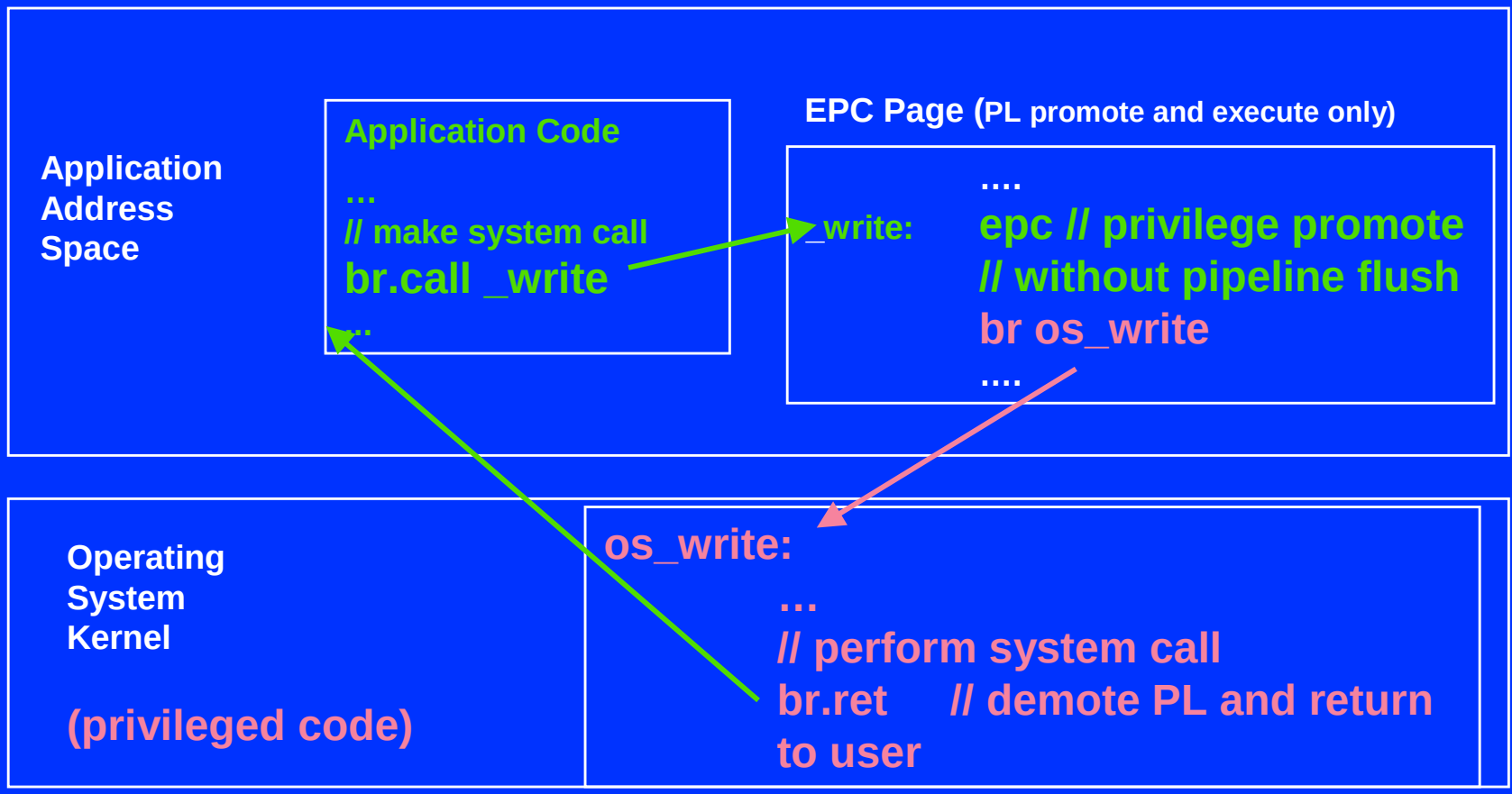
intel

Interruption Features

- Low interruption latency
  - Interruption delivery causes single pipeline break
  - Key state captured in on-chip registers
- State-save controlled by system software
  - Software makes performance/nesting trade-off
  - Shared mechanism for IA-64/IA-32 interruptions
- Efficient handler execution
  - Interruption vector table (IVT) contains code for interrupt service routine

Provides Fast And Flexible Interruptions For Large I/O Intensive Applications

Parallelism Across System Calls

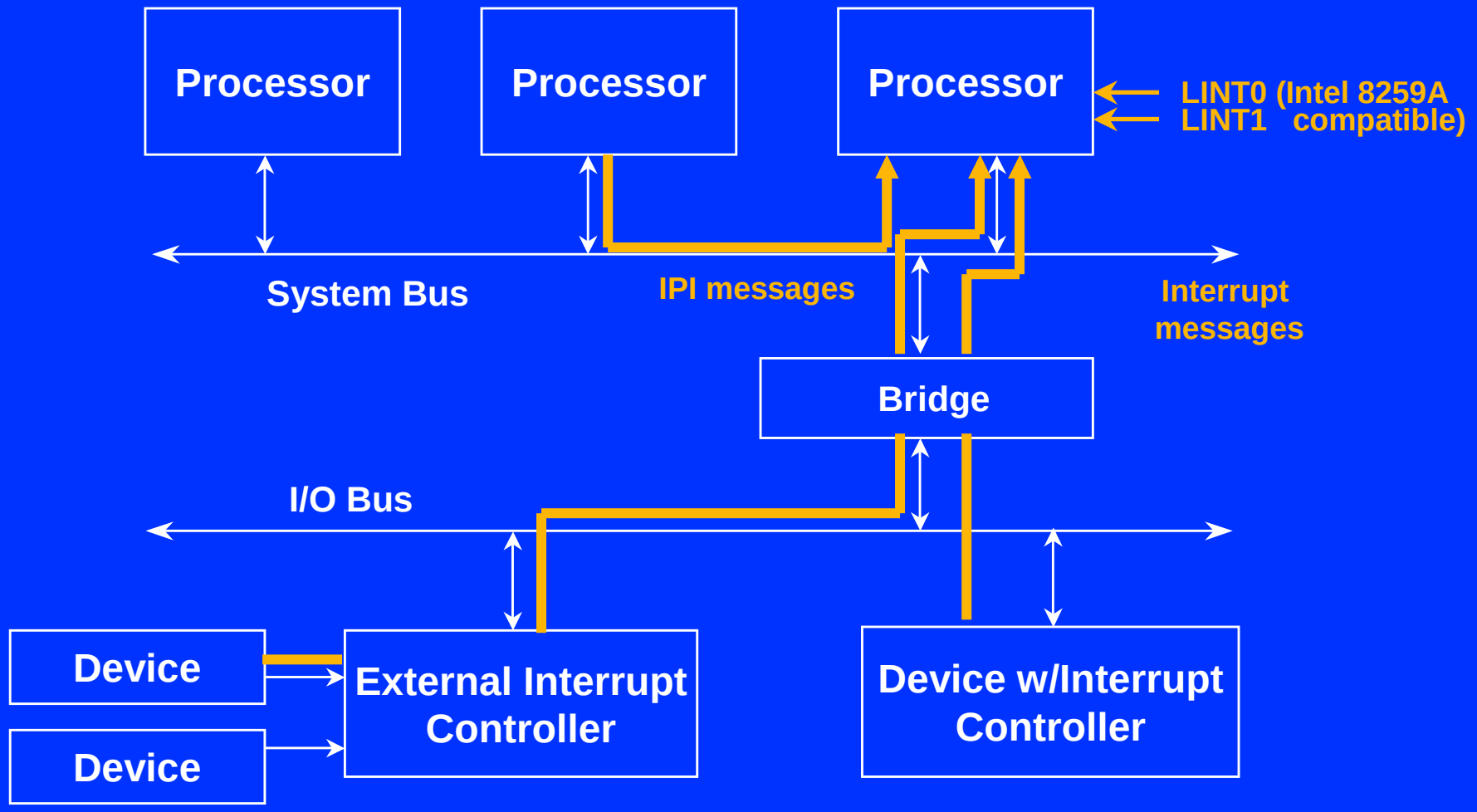


Fast System Calls Improve Synergy Between OS & Application

invent

intel

IA-64 External Interrupts



High Performance Message-Based Interrupts Compatible With Today's Platforms

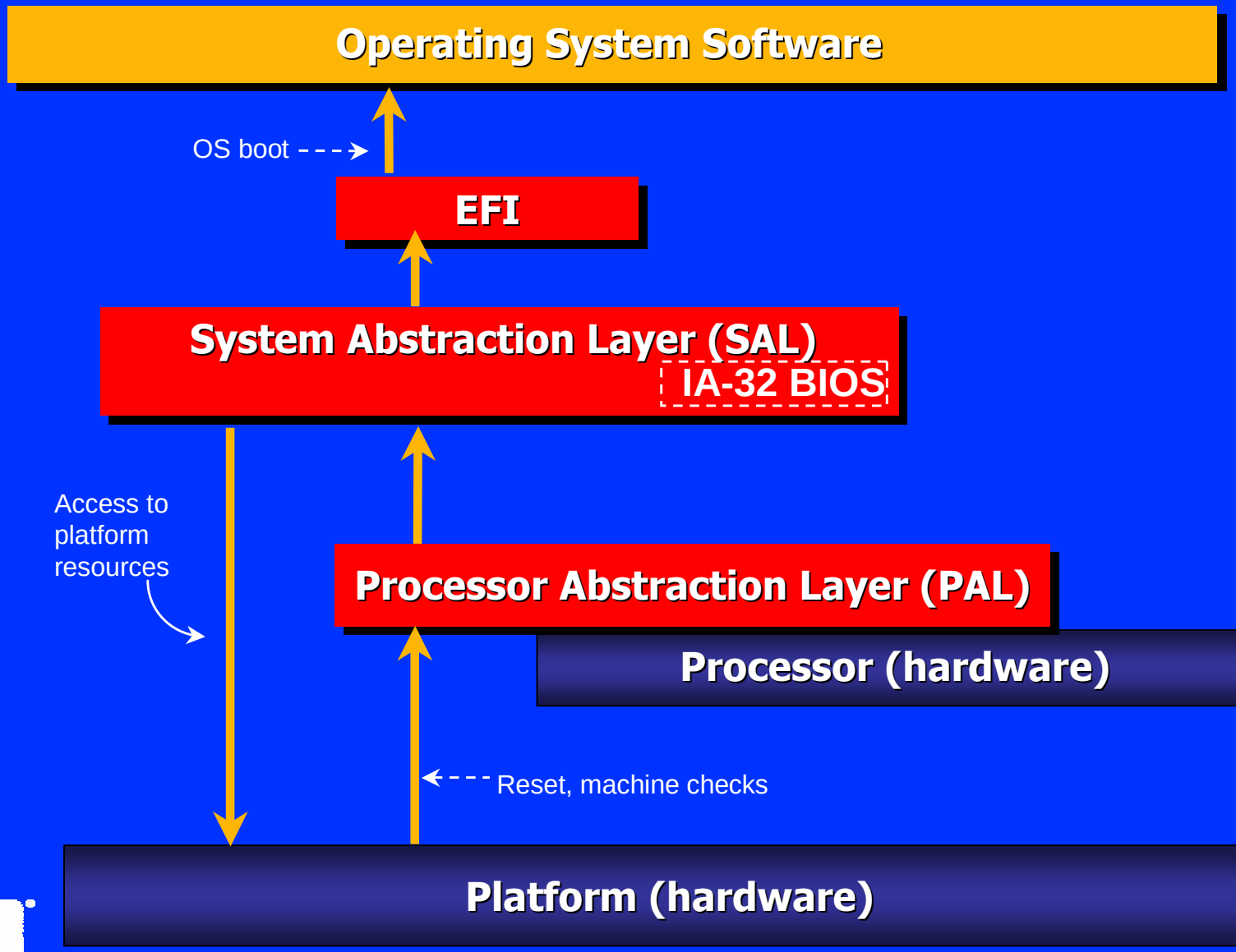
hp invent

intel

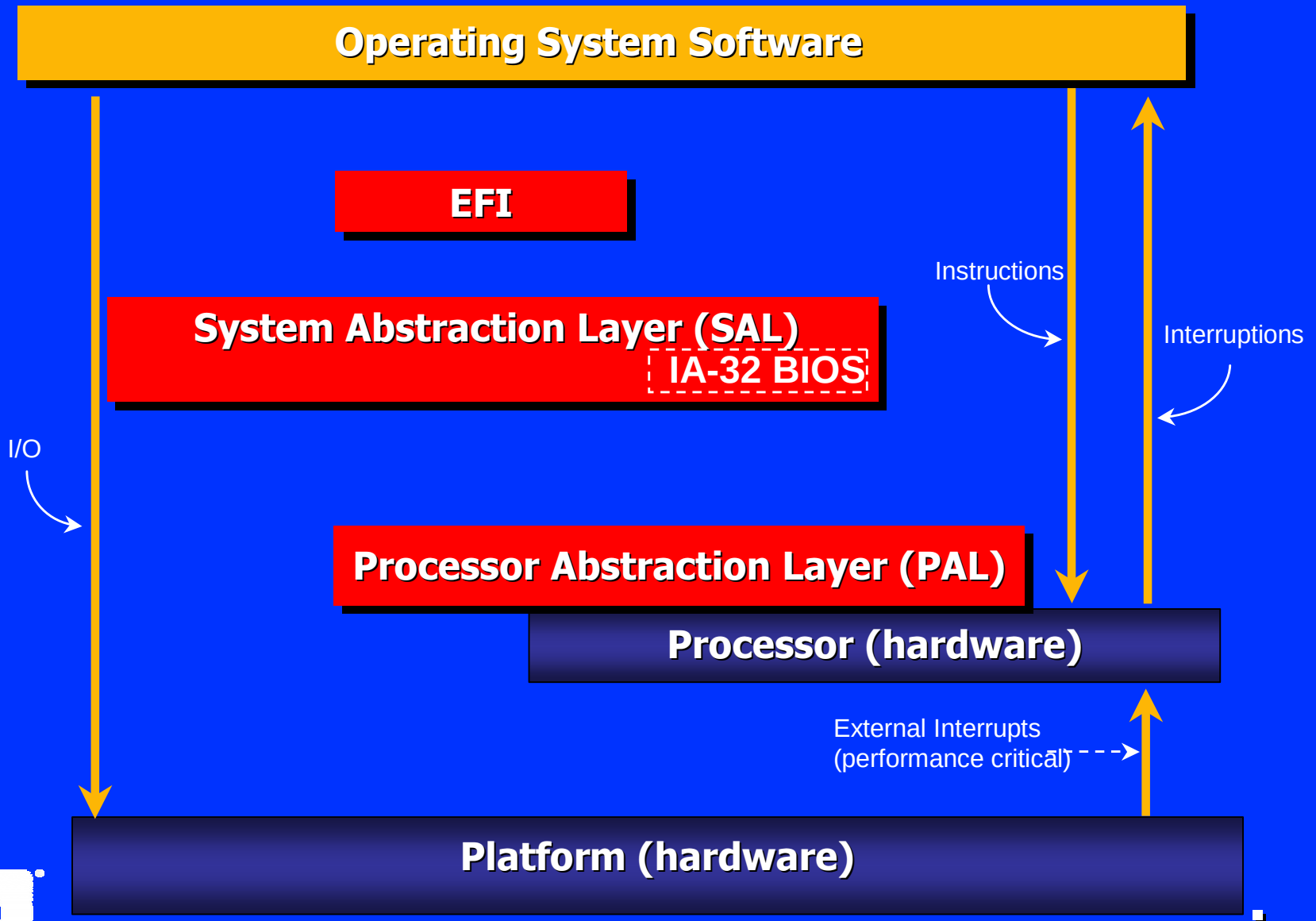
IA-64 System Architecture Agenda

- System Architecture Highlights
- Virtual Memory Model
- Interruption Model
- System Software Stack
- Reliability, Availability, Serviceability
- Parallelism & Scalability
- Compatibility
- Summary / Call To Action

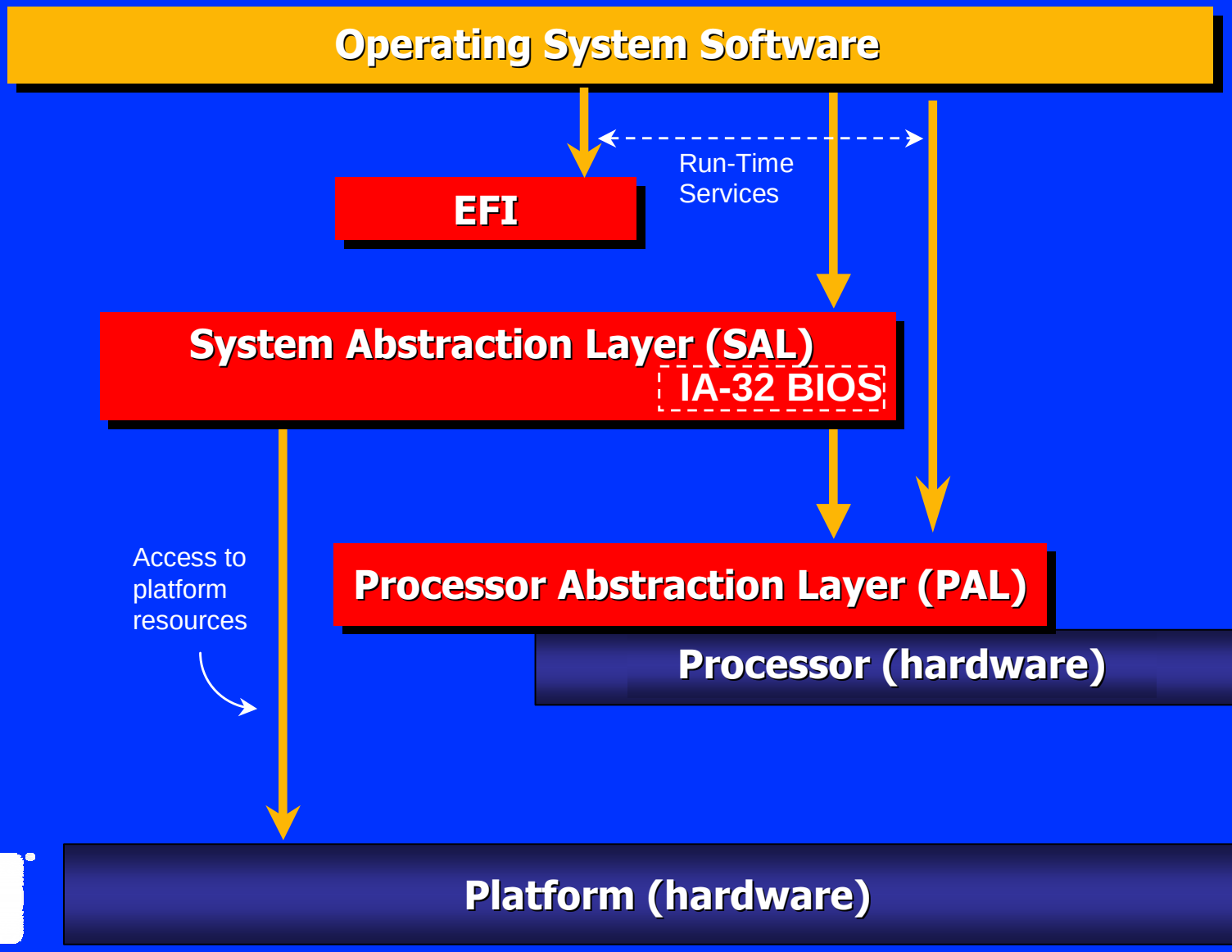
IA-64 System Software Stack: OS Boot



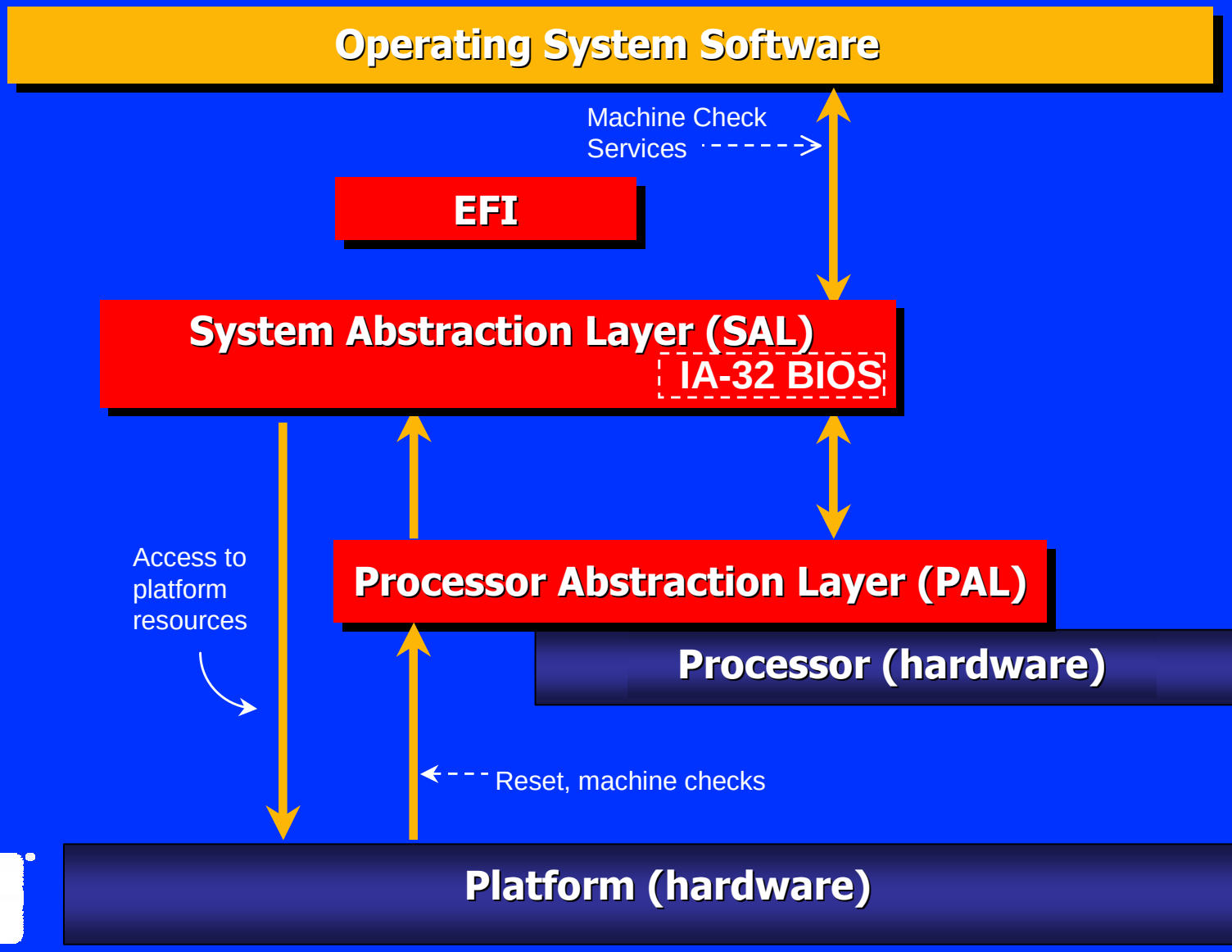
OS Running



OS Calls To Firmware Services



Machine Check Handling



Architected RAS Features

- Reliability
  - 3 levels of error signaling:
    - Continuable, local, and global
- Availability
  - Fine grained error containment by cooperation between hardware and firmware
- Serviceability
  - Extensive error logs for error analysis
  - Common error logs for firmware and OS

Advanced Machine Check Architecture For High Levels of Reliability, Availability, And Serviceability

IA-64 System Architecture Agenda

- System Architecture Highlights
- Virtual Memory Model
- Interruption Model
- System Software Stack
- Reliability, Availability, Serviceability
- Parallelism & Scalability
- Compatibility
- Summary / Call To Action

Parallelism and Scalability

- Excellent Multi-Processing Scalability
  - High performance relaxed ordering model
  - Flexible semaphore primitives
  - Hardware broadcast TLB purge
- OS Resource Scaling & Parallelism
  - Flexible # of PKRs, TRs, key & RID widths
  - Parallel update of control register writes (explicitly serialized)

IA-64 Features For Scalable MP Systems

Compatibility

- IA-64 supports PA-RISC & IA-32 Applications
- IA-64 supports IA-32 OS
  - Capable of running unmodified multi-processing IA-32 OS, e.g. NT4.0, Linux
- IA-64 OS supports IA-32 Platform peripherals
  - IA-64 support for legacy I/O port space
- Dependent upon OS & platform implementation

IA-64 Offers Full IA-32 Compatibility In Hardware: Platforms, OS, Applications

IA-64 System Architecture Agenda

- System Architecture Highlights
- Virtual Memory Model
- Interruption Model
- System Software Stack
- Reliability, Availability, Serviceability
- Parallelism & Scalability
- Compatibility
- Summary / Call To Action

IA-64 System Architecture Summary

Performance:

- Innovative architecture provides low overhead for major OS functions

Flexibility:

- Accommodates current OS & features for OS extensibility & evolution

Scalability:

- IA-64 platforms can run efficiently from UP to very large MP systems

Availablity:

- High levels of reliability & availability through enhanced RAS features

Interoperability:

- Well defined mechanisms & new firmware model ensures running of shrink-wrapped OS on variety of platforms

Compatibility:

- Provides efficient and transparent IA-32 compatibility at system level

Call to Action

- New detailed specifications available for download
  - OSVs & IHVs accelerate development of IA-64 system applications (device drivers, system debug tools, etc...)
- OSVs, IHVs, & ISVs take advantage of broad IA-64 enabling efforts to accelerate your porting efforts
- Take advantage of open source of IA-64 tools and Linux operating system

New Public IA-64 Docs

- **IA-64 Software Developer's Manual**
  - Info for system & application software, & development tools for IA-64
  - Software optimization techniques
  - Performance monitoring info for optimization support
- **More IA-64 Documentation:**
  - IA-64 Software Conventions and Runtime Architecture Guide
  - Assembly Language Reference Guide
  - IA-64 assembler & reference guide
  - IA-64 Processor-specific Application Binary Interface
  - System Abstraction Layer Specification

  and more ...

IA-64 Docs Available On Internet:

www.hp.com/go/ia-64/

developer.intel.com/design/ia-64/devinfo.htm


Intel
Developer
Forum
Spring 2000
intel