“Book E”: An Enhanced PowerPC Architecture

Thomas Sartorius
IBM Microelectronics
Embedded PowerPC Principal Architect

Pete Wilson
Motorola Semiconductor Products Sector
Senior Member of the Technical Staff

Presented by:
Thomas Sartorius

What is “Book E”?

Book E is the project name for the collaborative effort between Motorola and IBM to enhance the PowerPC Architecture™

Book E is a complete architectural definition, not just an addendum to the PowerPC Architecture

Why are Motorola and IBM defining Book E?

PowerPC

The marketplace is moving to application-specific systems with diverse needs

- **Book E objectives:**
  - Provide an enhanced architecture definition with 64-bit capabilities
  - Increase architectural flexibility
  - Deliver optimizations specific to embedded systems

MOTOROLA

IBM

Why are Motorola and IBM defining Book E?

PowerPC

Customers have identified the need to maintain the PowerPC Architecture, now and into the future

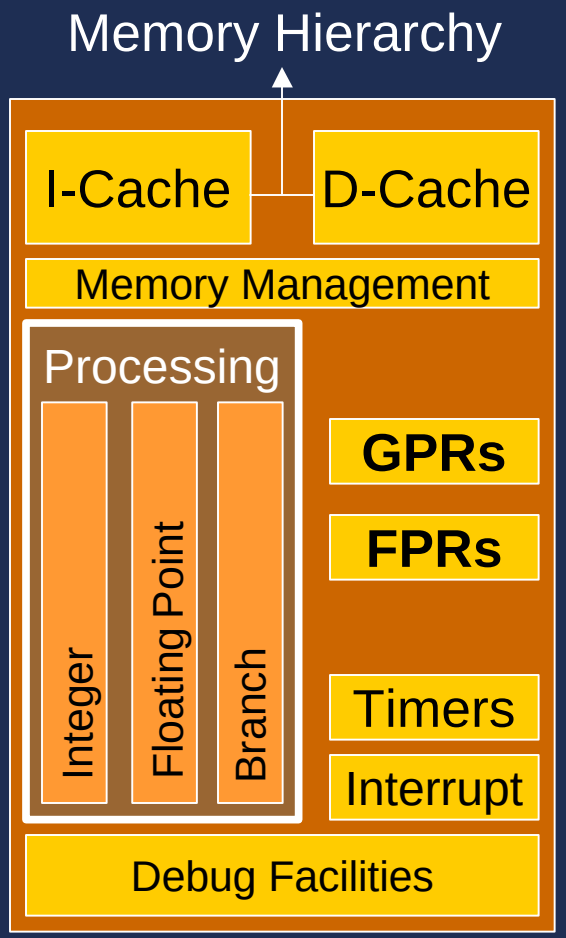
- **Book E objectives:**
  - Eliminate non-substantive architectural differences in future IBM and Motorola embedded PowerPC implementations
  - Provide compatibility for existing PowerPC applications

MOTOROLA

IBM

Complete 64-bit architecture

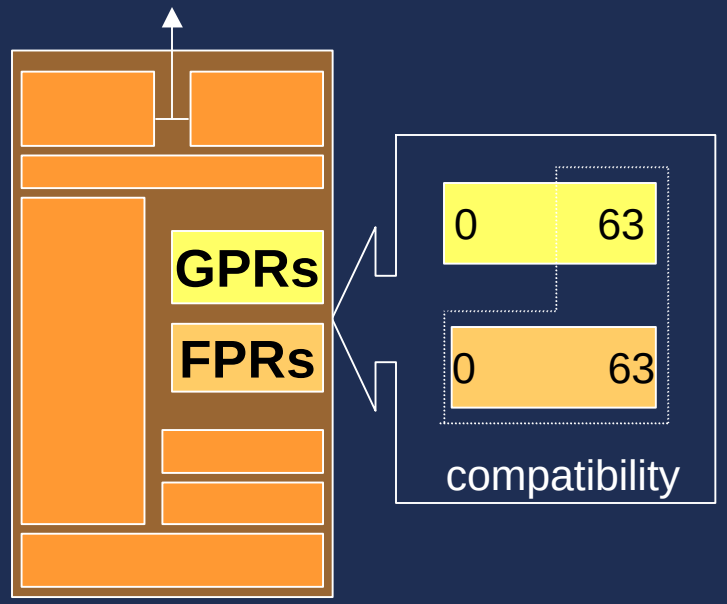
- **64-bit addressing required for high-performance apps (e.g. RAID, networking, telecom)**
- **64-bit Book E processors can execute 32- and 64-bit applications**
  - Supports 32-bit legacy code
- **Book E allows 32-bit or 64-bit processor implementations**
  - Optimize cost and performance for diverse system requirements



Unified 32-/64-bit architecture, without modes

PowerPC

- Existing load/store/branch instructions calculate 32-bit addresses
- Existing arithmetic instructions produce 32- and 64-bit results
- 84 new instructions for 64-bit addressing and extended arithmetic
- Unified 32-/64-bit MMU
  - Single MMU covers both 32- and 64-bit address translation
  - Easier OS migration

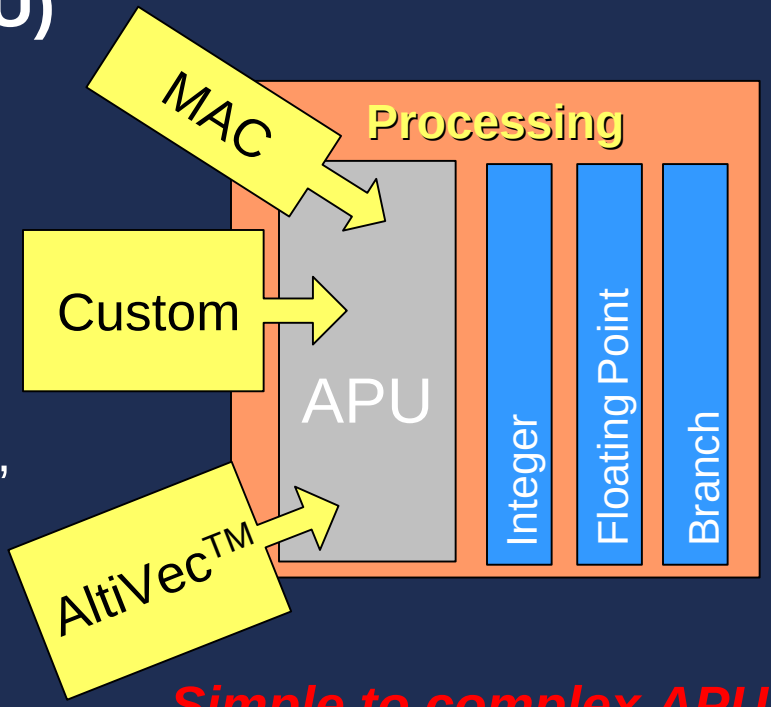


MOTOROLA

IBM

Powerful capability to implement custom functionality

- **Book E allows Application-Specific Processing Units (APU) for instruction set extensions**
  - Single instruction stream
  - Examples:
    - DSP
    - Multimedia
    - Custom (e.g. Servo, print engine, automotive control)
- **An APU design can be:**
  - Simple, sharing use of GPR file
  - Complex, separate AP register file with direct load/store access

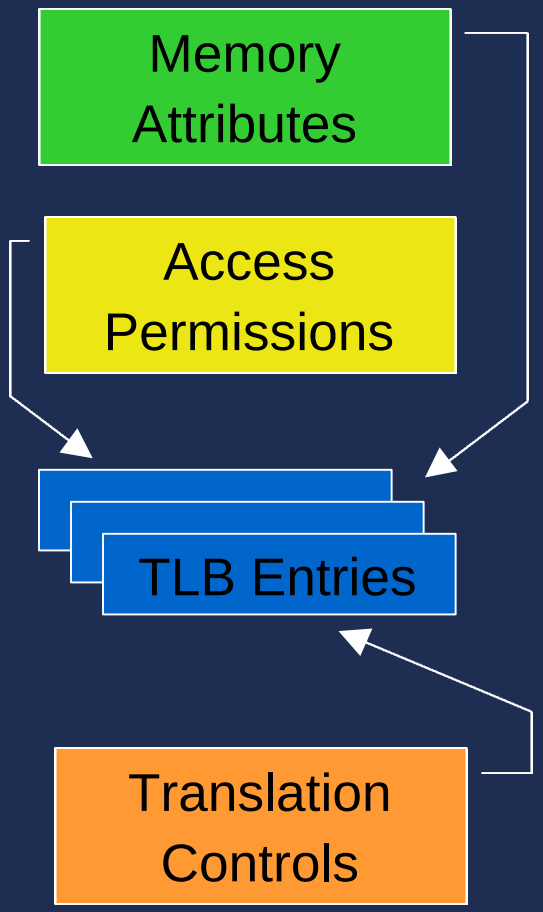


Simple to complex APUs

Adaptable memory management for diverse operating environments

PowerPC

- **Architected software-managed TLB**
  - Architecture defines TLB functions
  - System defines page table size, location, organization, supported page sizes, and TLB replacement strategy
- **Variable page sizes, 1KB to 1TB**
  - More efficient memory allocation
  - Compact TLB can map large memory
- **Memory attributes**
  - Cache policy
  - Endianness
  - User-definable (e.g., code compression, memory hierarchy control, etc.)

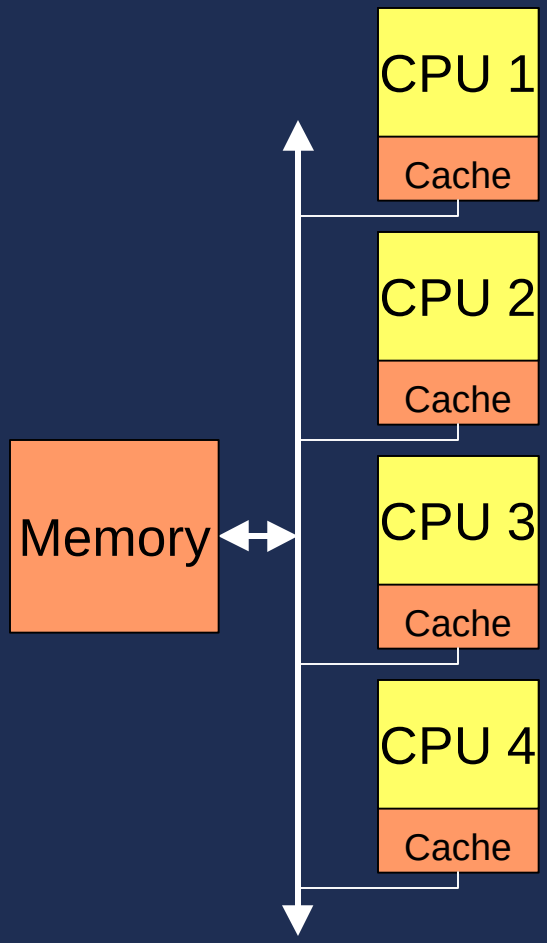


MOTOROLA

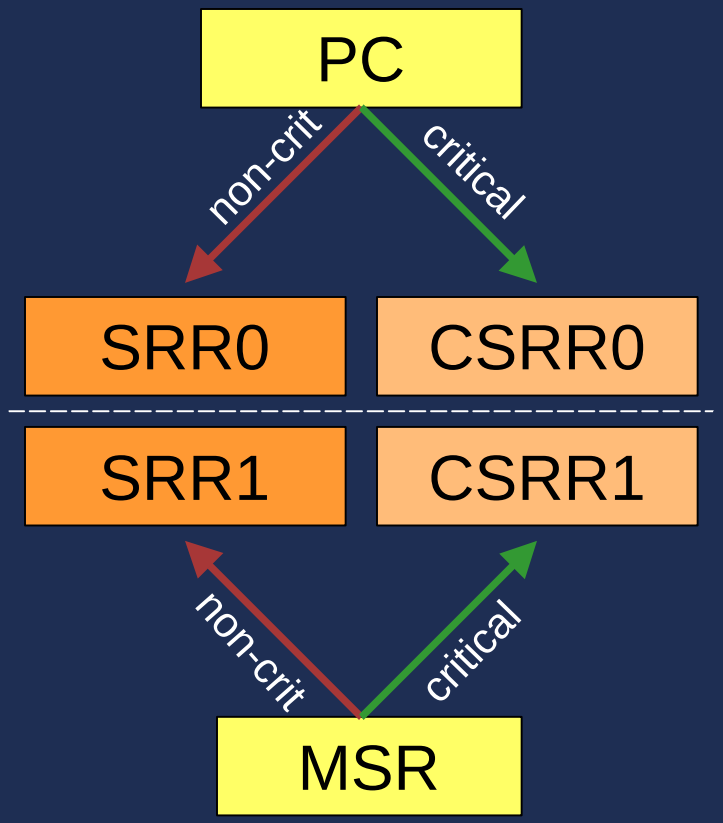
IBM

High-performance memory architecture

- **Maintains PowerPC memory model**
  - Weakly-ordered memory access
  - Full MP cache coherent architecture
    - Implementation freedoms remain
- **Improved memory barrier mechanisms**
  - Speeds lock management
  - Benefits heavily multi-threaded apps like Java
- **Support for cache locking**
  - Architected exception types and vectors for lock-related errors

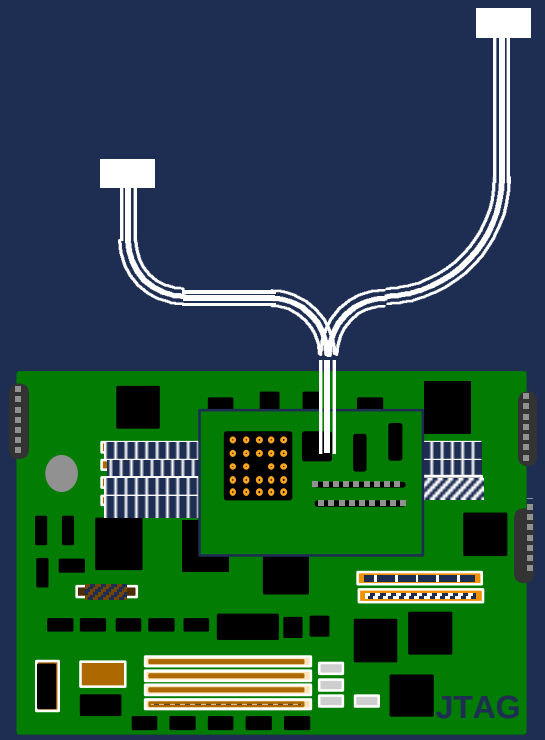


Two interrupt priority levels for real-time systems



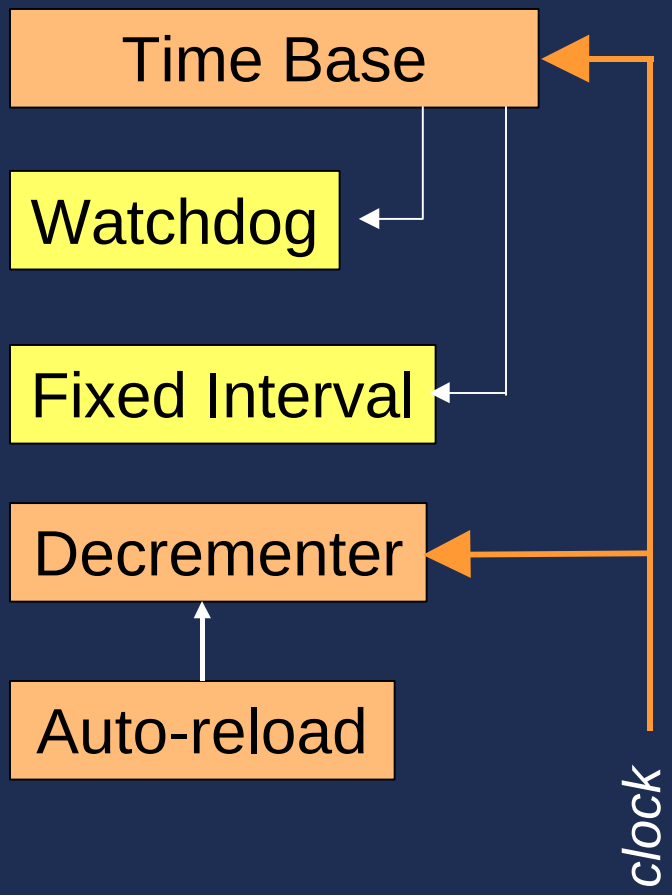
- **Critical interrupt level for critical input, debug, watchdog**
  - Low latency response to high-priority devices
  - Robust software debug of non-critical handlers
  - Improved system reliability
- **Non-critical interrupt level for TLB miss, memory protection, illegal op-codes, etc.**

Robust debug for high-performance systems



- **Architected debug framework for application developers and tool vendors**
  - Instruction address compare
    - Exact match, address bit mask, inclusive/exclusive ranges
  - Data address compare
    - Including data value compare
  - Other debug events
    - Instruction discontinuity, single step, asynchronous
- **Architecture permits subset of debug facilities**

Expanded timer facilities for hard real-time constraints



- **64-bit time base register, for general system time maintenance**
- **Improved 32-bit decrementer**
  - Auto-reload and auto-stop modes
  - SW timer for multitasking
- **Fixed interval timer, for periodic system maintenance**
- **Watchdog timer, for system error recovery**

Architectural compatibility

- **Backward and forward compatibility with existing PowerPC implementations**
  - Previous PowerPC 32-bit application binaries will run unmodified on new Book E implementations
  - New Book E 32-bit application binaries will run unmodified on previous PowerPC implementations
- **Improvements for system software**
  - Significant enhancements for RTOS memory management
  - More robust interrupt handling
- **Consistent architecture for application and tool developers**

Commitment to Book E

- **Motorola and IBM:**
  - Have jointly defined the Book E enhanced PowerPC Architecture
  - Will continue to collaborate to enhance the architecture
  - Will independently develop products based on the Book E architecture
  - Each have the ability to sub-license the architecture
- **The Book E specification is available today at:**
  - http://motorola.com/PowerPC
  - http://www.chips.ibm.com/products/powerpc

Summary

- Book E provides enhancements to address future marketplace requirements, while maintaining support for existing PowerPC-based systems
- Book E provides flexibility for application-specific optimizations while providing architectural consistency for software development
- IBM and Motorola are making Book E a premier architecture for high-performance and cost-effective applications