ADVANCED PROGRAMMABLE INTERRUPT CONTROLLER
(APIC)

Architecture

PK Nizar
Intel Corporation

AGENDA

* Limitations of Current Interrupt Management Solutions

* APIC Architecture Overview

* APIC Features Benefits Summary
LIMITATIONS OF CURRENT INTERRUPT SOLUTIONS

Interrupt Controller Access (SPL()) by OS is Major Hot Spot

LIMITATIONS OF 8259A INCREASE S/W OVERHEAD

* Priority / IRQ Bindings Fixed
* One Interrupt per Priority
* S/W Workaround Prioritization by Early EOI, Explicit masking
* Increased Overhead
* Slow Access in I/O Space

- Mask Updates -------------- 5.3 usec
- Interrupt Vector Fetch ---- 3 usec
- EOI Cycles ---------------- 3.1 usec
- Timing Loops --------------- 5 - 10 usec

APIC ARCHITECTURE OVERVIEW
APIC ARCHITECTURE SUMMARY

* Distributed Architecture
* Dedicated Interrupt Communications Bus
* Programmable Interrupt Priority Model
* High Performance Priority Management
* MP Interrupt Management

DISTRIBUTED ARCHITECTURE

PROCESSOR

APIC LOCAL UNIT

APIC BUS

I/O SUBSYSTEM

LINT0, LINT1

Interrupts

* Masking/Prioritizing
* Pending/Servicing
* Routing/Distribution
* Delivery Semantics
* Interrupt Identification
APIC BUS

**FEATURES:**
* Local Units and I/O Units Communicate via APIC Bus
* Error Handling
* Distributed Arbitration

**BENEFITS:**
* APIC Bus allows Scalability in MP System
* Improved System Bus Utilization

I/O Interrupts → APIC I/O Unit → APIC BUS → APIC Local Unit → PROCESSOR

APIC Bus allows Scalability in MP System

**PROGRAMMABLE INTERRUPT PRIORITY MODEL**

**Vector Identifies Interrupt:**
* Identifies Interrupt Source
* Allows Destination Processor to Find Handler
* Programmable Interrupt Source to Vector Mapping
* Maximum 256 Vectors (x86 Interrupt Table Size)

**Interrupt Priority Implied by Vector:**
* Priority = Vector / 16
* 16 interrupt Priorities
* 16 Interrupts per Priority Level
HIGH PERFORMANCE PRIORITY MANAGEMENT

* APIC MAINTAINS TWO PROCESSOR PRIORITY INDICATIONS

  ** TASK PRIORITY**
  * Tracks OS Defined Task/Process/Thread Priority
  * Updated by Writes to Local APIC's Task Priority Register
    - Task Switch
    - Task Priority Changes (e.g. SPL (\))

  ** INTERRUPT HANDLER PRIORITY**
  * Equal to Priority of Highest In-Service Interrupt
  * Handler Can Explicitly Raise/Restore Priority via Task Priority Register
  * APIC Tracks Handler Priority Across Interrupt Nesting/Unnesting

* CURRENT PROCESSOR PRIORITY IS MAXIMUM OF THESE TWO

HIGH PERFORMANCE PRIORITY MANAGEMENT

* APIC masks all Interrupts Lower or Equal to the Current Priority Level

* Memory Mapped Task Priority Register

* Useful for Synchronized Access of Shared Resources in OS

* Provides Mutual Exclusion

* Faster SPL () / R/LQL () Routines
MP INTERRUPT MANAGEMENT

* Fully Symmetric, Static/Dynamic Interrupt Distribution
* Group Broadcast, Fixed or Lowest Priority
* Focus Processor concept
* Logical or Physical Addressing
* Flexible Inter-Processor interrupts
* Supports Tasks Migration and Interrupt forwarding

APIC FEATURES BENEFITS SUMMARY

<table>
<thead>
<tr>
<th>Features</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmable Interrupt Priority</td>
<td>OS Customizable Priority Model</td>
</tr>
<tr>
<td>CPU/Task Priority Tracking</td>
<td>Reduces Software Overhead</td>
</tr>
<tr>
<td>32-bit Memory Mapped Registers</td>
<td>Faster SPL( ) Routines</td>
</tr>
<tr>
<td>APIC Bus</td>
<td>Posted Writes; Performance Boost</td>
</tr>
<tr>
<td>Dynamic Interrupt Distribution</td>
<td>Scalability in MP Systems</td>
</tr>
<tr>
<td>Focus Processor</td>
<td>Load Balancing in MP systems</td>
</tr>
<tr>
<td></td>
<td>Improves Interrupt Latency</td>
</tr>
</tbody>
</table>