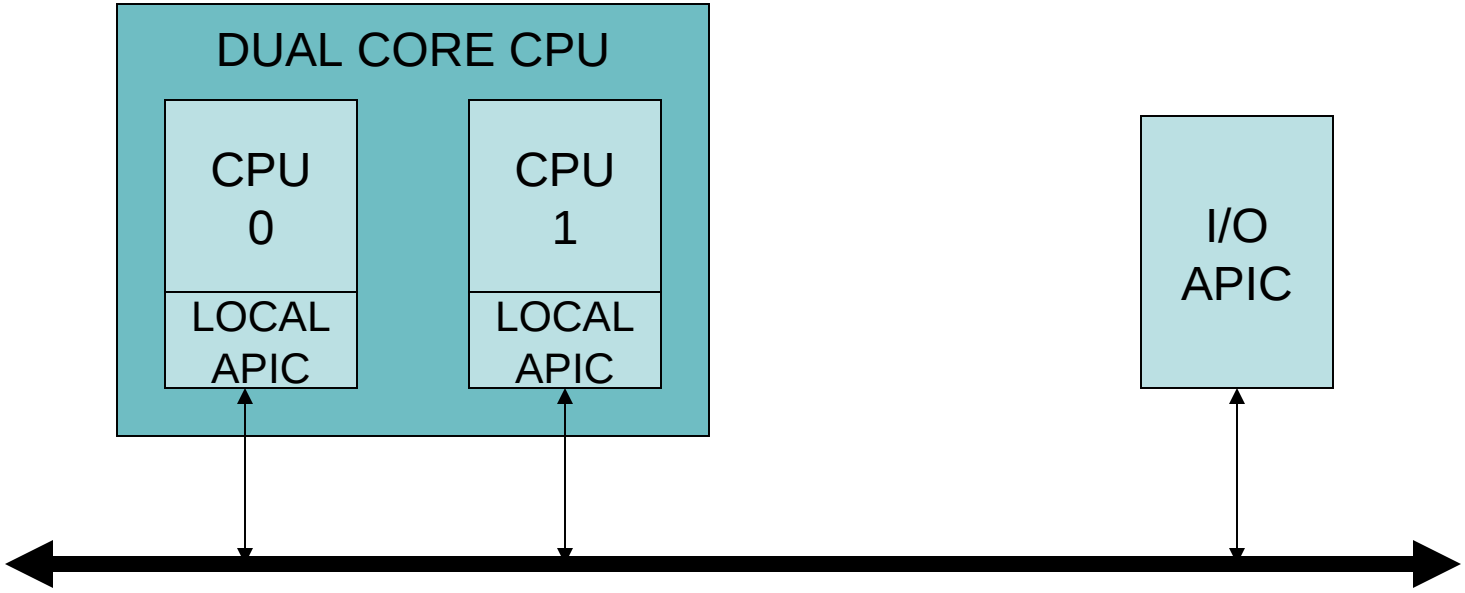
Advanced Programmable Interrupt Controllers

Multiprocessor-systems require enhanced circuitry for signaling of external interrupt-requests

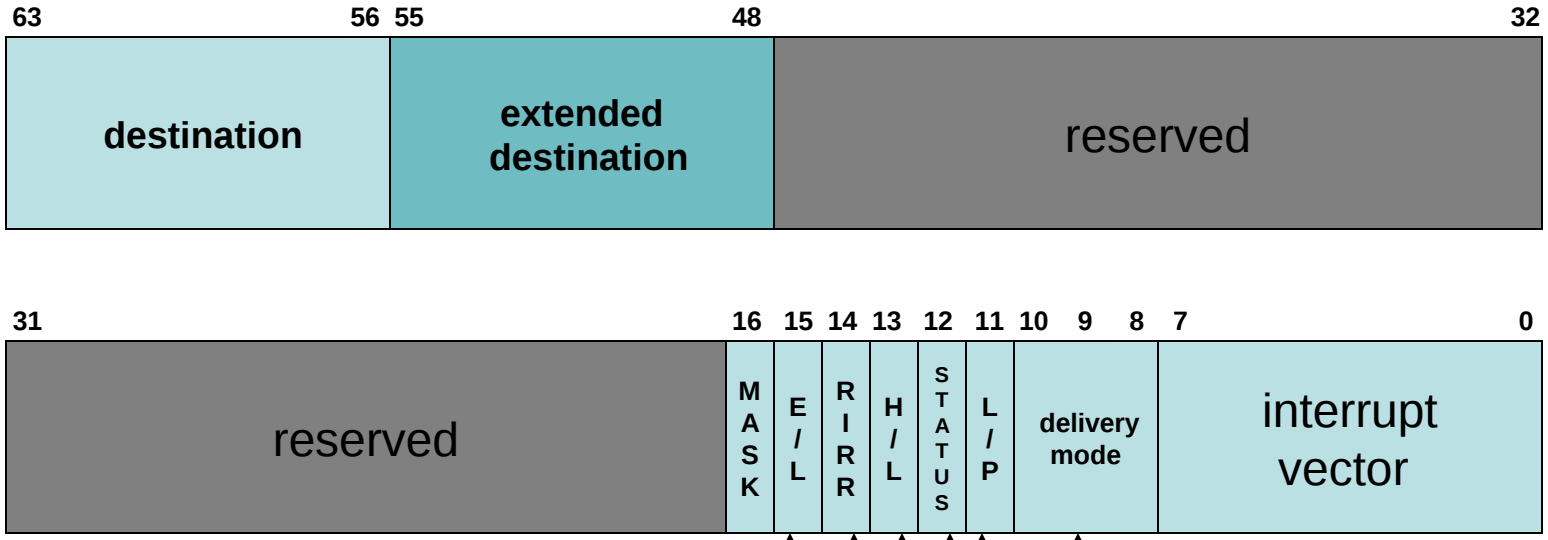
Multiple Logical Processors



Advanced Programmable Interrupt Controller is needed to perform ‘routing’ of I/O requests from peripherals to CPUs

(The legacy PICs are masked when the APICs are enabled)

Redirection Table Entry



Trigger-Mode (1=Edge-triggered, 0=Level-triggered)

Remote IRR (for Level-Triggered only)
0 = Reset when EOI received from Local-APIC
1 = Set when Local-APICs accept Level-Interrupt sent by IO-APIC

Interrupt Input-pin Polarity (1=Active-High, 0=Active-Low)

Delivery-Status (1=Pending, 0=Idle)

000 = Fixed
001 = Lowest Priority
010 = SMI
011 = (reserved)
100 = NMI
101 = INIT
110 = (reserved)
111 = ExtINT

Destination-Mode (1=Logical, 0=Physical)

I/O APIC Documentation

"Intel I/O Controller Hub (ICH7) Family Datasheet"

available online at

http://www.intel.com/design/chipsets/datashts/307013.htm

Our ‘ioapic.c’ kernel-module

- This Linux module creates a pseudo-file (named ‘/proc/ioapic’) which lets users view the current contents of the I/O APIC Redirection-Table registers
- You can compile and install this module for our classroom and CS Lab machines or our Core-2 Duo (“anchor”) machines

Our ‘anchor’ systems

Mapping of IRQ-lines to Interrupt-ID numbers

0 (masked)
1 (keyboard) → 0x39
2 (timer) → 0x31
3 ( ) → 0x41
4 (serial-uart) → 0x49
5 ( ) → 0x51
6 (diskette-controller) → 0x59
7 (parallel-port) → 0x61
8 (real-time-clock) → 0x69
9 (acpi) → 0x71
A ( ) → 0x79
B ( ) → 0x81
C (mouse) → 0x89
D ( ) → 0x91
E (hard-disk) → 0x99
F ( ) → 0xA1
10 (ethernet) → 0xA9
11 ( ) → 0xB1
12 ( ) → 0xB9
13 ( ) → 0xC1
14 (masked)
15 (masked)
16 (masked)
17 ( ) → 0xC9