Interrupt Mask

* There might be situation that processor doesnot want to entertain the interrupt raised by devices. Situation when we need atomicity in code.

* Single bit, hardware support.
  - PIC → Programmable Interrupt controller
  - APIC → Advanced " " "
    - → handles interrupts
    - → lines between controller & devices
    - → controller communicates with CPU to handle interrupts.
    - → Before it was put as a hardware chip on the computer
    - → These days it is incorporated with CPU.
    - → XINU provides us way to code this PCI.

* what is inter-processor interrupt ?

There could be a way that one processor interrupt other processor. CPU interrupting other CPU. Applications using multiple CPUS. Very expensive. When one CPU is changing the page-mapping, it has to interrupt other CPU to immediately stop because CPU needs to update its cache.

Interrupt Processing

* Vector for all interrupts.
* Interrupt handler for each interrupt.
* OS has to configure these.
* Interrupt handler will use registers to save data during its execution and also restore after it returns.
* We need to call a special instruction (IRET) after interrupt returns.

restore old program state

interrupt mask

instruction pointer (PC)

* Hardware returns to location where interrupt occurred