

# STOS - Events

Gabriel Laskar <[gabriel@lse.epita.fr](mailto:gabriel@lse.epita.fr)>

# Outline

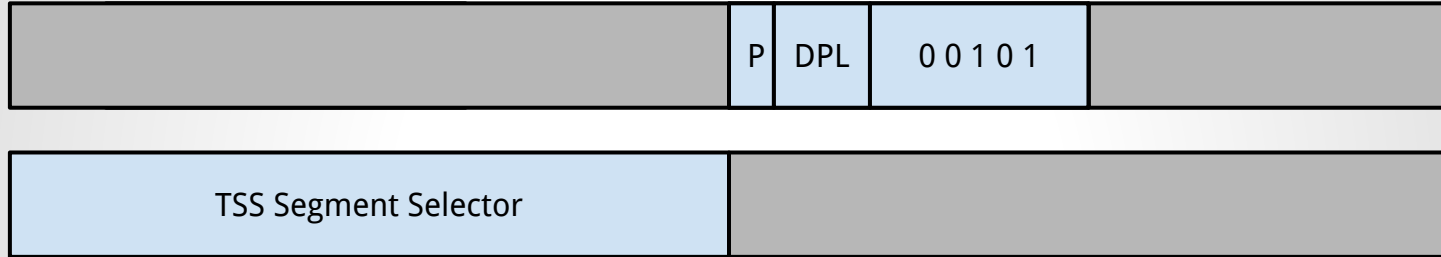
- Interrupts and events
- IDT
- IRQ
- PIC
- How to test : keyboard, mouse
- PIT

# Interrupt and Exception Handling

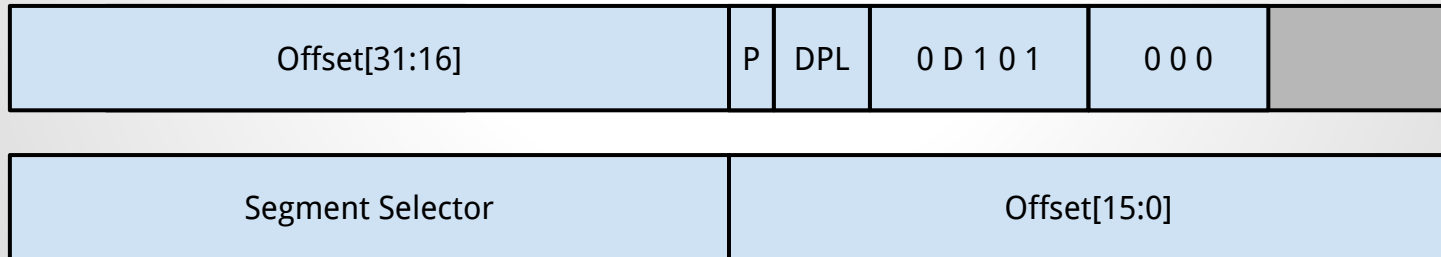
- Exception: Synchronous with program execution (division by zero, access to invalid address)
- Interrupt: Asynchronous with program execution. Generated by devices external to the CPU

# Interrupt Descriptor Table

## Task Gate



## Interrupt Gate



# Load a new IDT

```
struct idtr idtr = {  
    .base = idt,  
    .limit = sizeof(idt) - 1  
};  
  
__asm ("lidt %0\n" : : "m" (idtr):  
"memory");
```

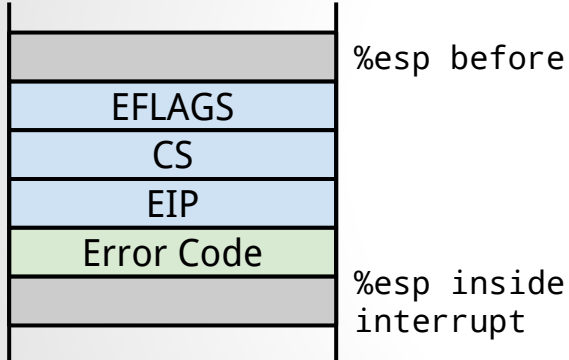
# Exceptions on x86

0	Divide by 0
1	Debug
2	NMI
3	Breakpoint
4	Overflow
5	Bound Range Exceeded
6	Invalid Opcode
7	Device not Available

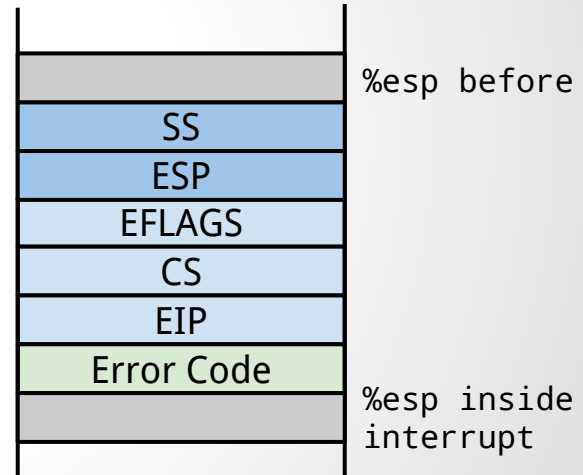
8	Double Fault
9	Coprocessor
10	Invalid TSS
11	Segment not present
12	Stack segment fault
13	General Protection Fault
14	Page Fault
15-31	Reserved by Intel

# Context Switching

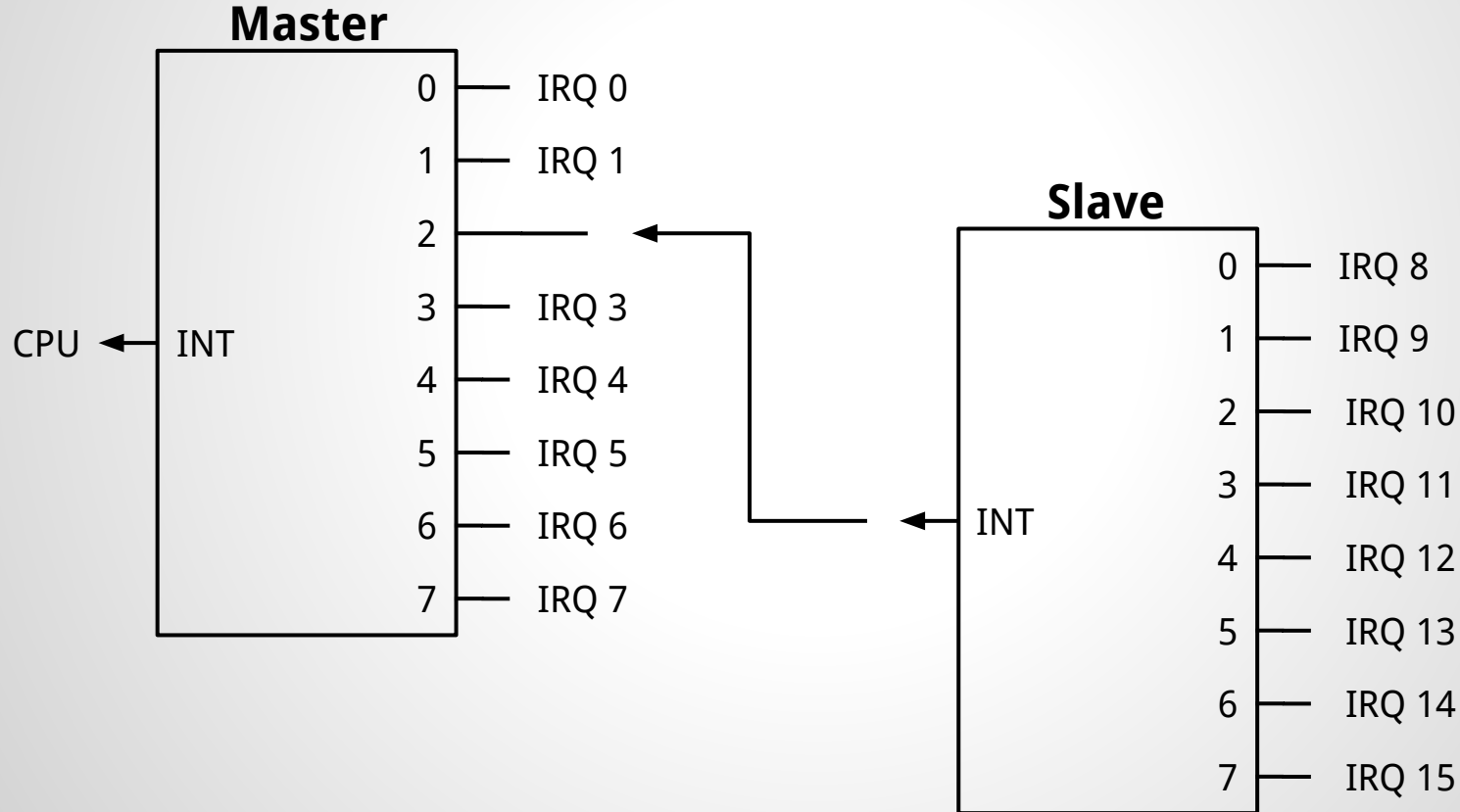
Stack with no privilege change



Stack with privilege change

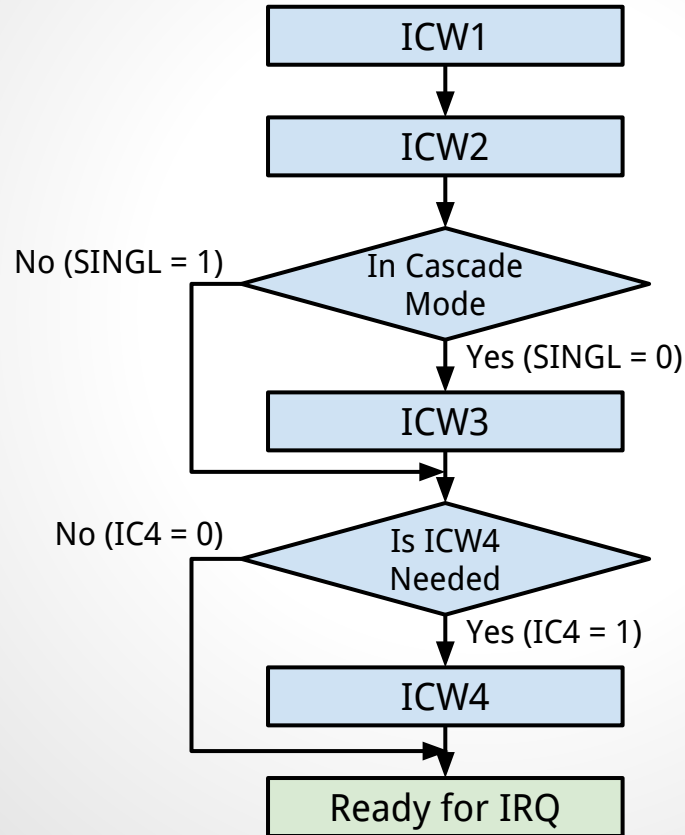


# Programmable Interrupt Controller





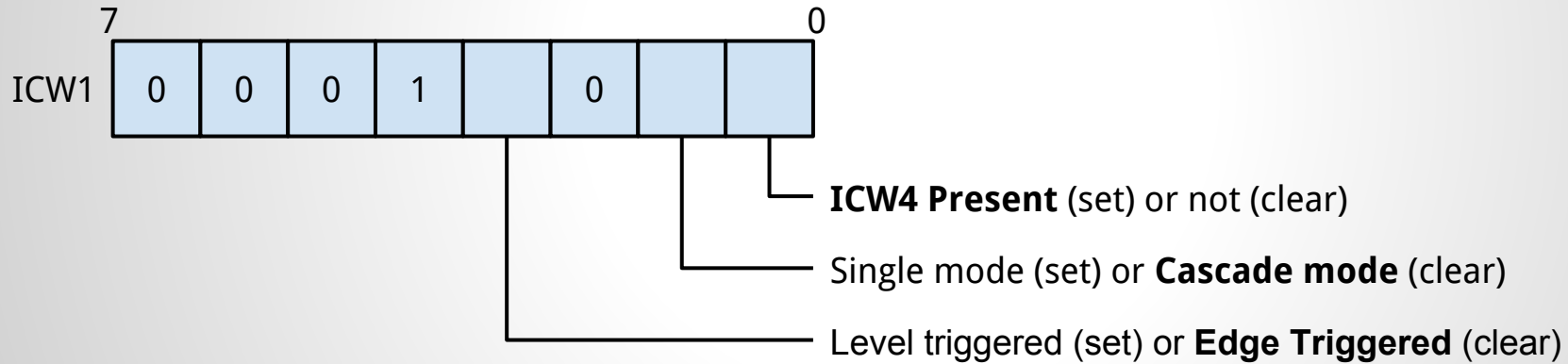
# PIC Initialization



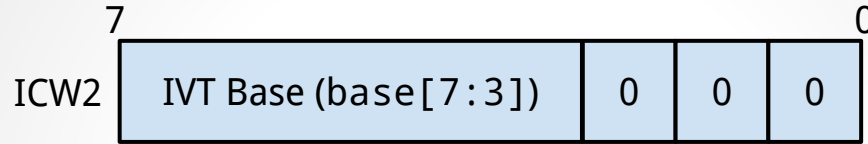
# PIC Ports

- 0x20: master PIC port A
- 0x21: master PIC port B
- 0xA0: slave PIC port A
- 0xA1: slave PIC port B
- ICW1: port A
- ICW2: port B
- ICW3: port B
- ICW4: port B
- OCW1: port B
- OCW2: port A

# ICW1

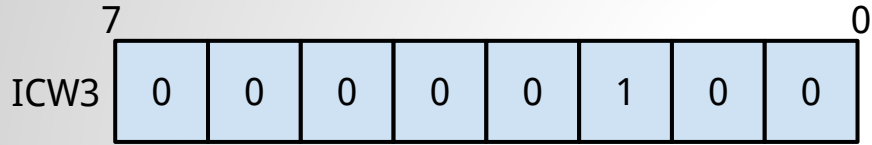


# ICW2

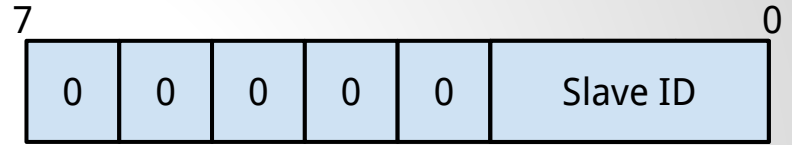


IVT Base: IDT entry aligned to 8 bytes

# ICW3

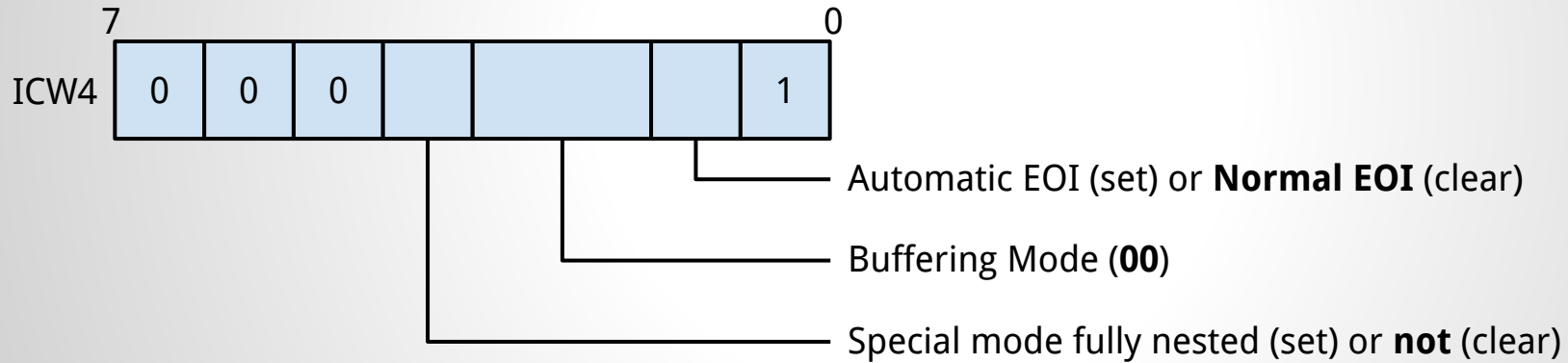


Master: For each bit, indicate whether a slave PIC is connected or not

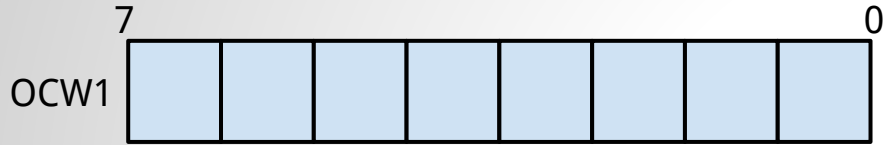


Slave: Indicate to the slave his slave ID (which pin of the master it is connected to)

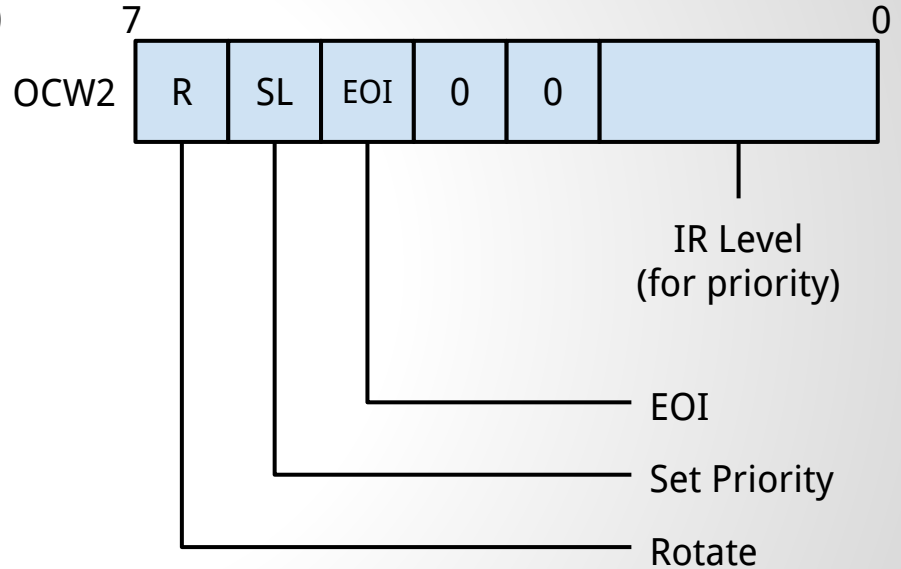
# ICW4



# OCW1 & OCW2



- OCW1 describe the interrupt mask
- We need to send a non-specific End of Interrupt



# Typical wiring of the PIC

- IRQ0: PIT
- IRQ1: Keyboard
- IRQ2: Slave
- IRQ3: COM2/COM4
- IRQ4: COM1/COM3
- IRQ5: HD or LPT2
- IRQ6: Floppy
- IRQ7: LPT1 or Spurious
- IRQ8: RTC
- IRQ9: PCI (PIRQA, PIRQD)
- IRQ10: PCI (PIRQB)
- IRQ11: PCI (PIRQC)
- IRQ12: PS/2 mouse
- IRQ13: Math coprocessor
- IRQ14: HD1
- IRQ15: HD2 or Spurious



# Methodology

- Write IDT management functions
  - allocate/clean IDT
  - set an interrupt gate in the IDT
- Write the context saving/restoring routines in assembly code
- Implement the exceptions and interrupts wrappers
- Test interrupts with simple handlers

# Methodology

- Initialize the PIC
  - send ICWs to both master and slave PICs
  - mask all interrupts
- Write very simple debug handlers like keyboard, mouse or RTC
- Do not forget to enable hardware interrupts using sti

# Testing: Keyboard & Mouse

- Simple devices (no setup, 1 IRQ)
- keyboard for master
- mouse/RTC for slave
- see K course and website for that (or osdev)

# Programmable Interval Timer

- 3 counters:
  - **Counter 0:** irq at user-defined frequency
  - Counter 1: used for DRAM refresh
  - Counter 2: used for speaker
- 6 modes:
  - Mode 0: Interrupt on terminal count
  - Mode 1: hardware retriggerable one-shot
  - **Mode 2:** rate generator
  - Mode 3: square generator
  - Mode 4: Software Triggered Strobe
  - Mode 5: Hardware Triggered Strobe

# PIT Configuration

## Registers:

- 0x40: **Counter 0**
- 0x41: Counter 1
- 0x42: Counter 2
- 0x43: Control Register

## Divider:

- write divider in counter register  
divider = base / desired
- base: 1193182 Hz

