

# CSE121: IoT

---

## Timer/Interrupt

Jose Renau  
[renau@ucsc.edu](mailto:renau@ucsc.edu)



**Baskin School of Engineering**  
University of California, Santa Cruz

# Announcements

---

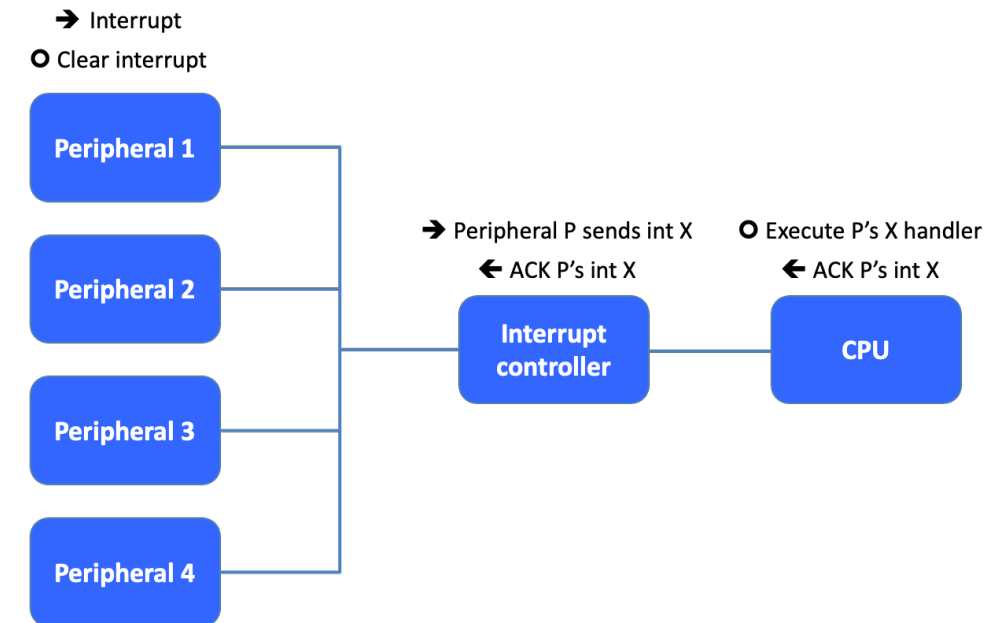
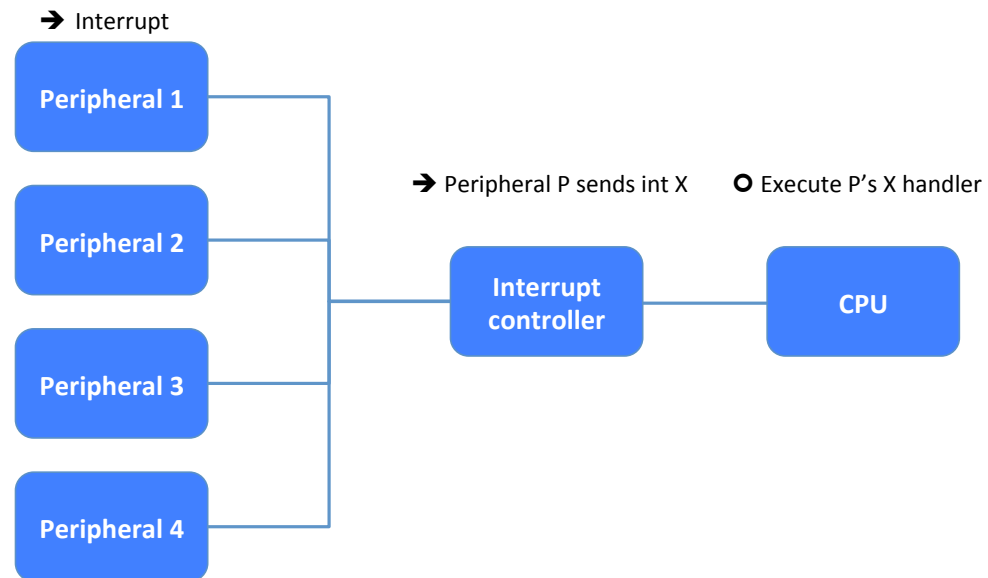
- Lab3 this week
- Lab4 next week

-

- <https://www.freertos.org/a00106.html>

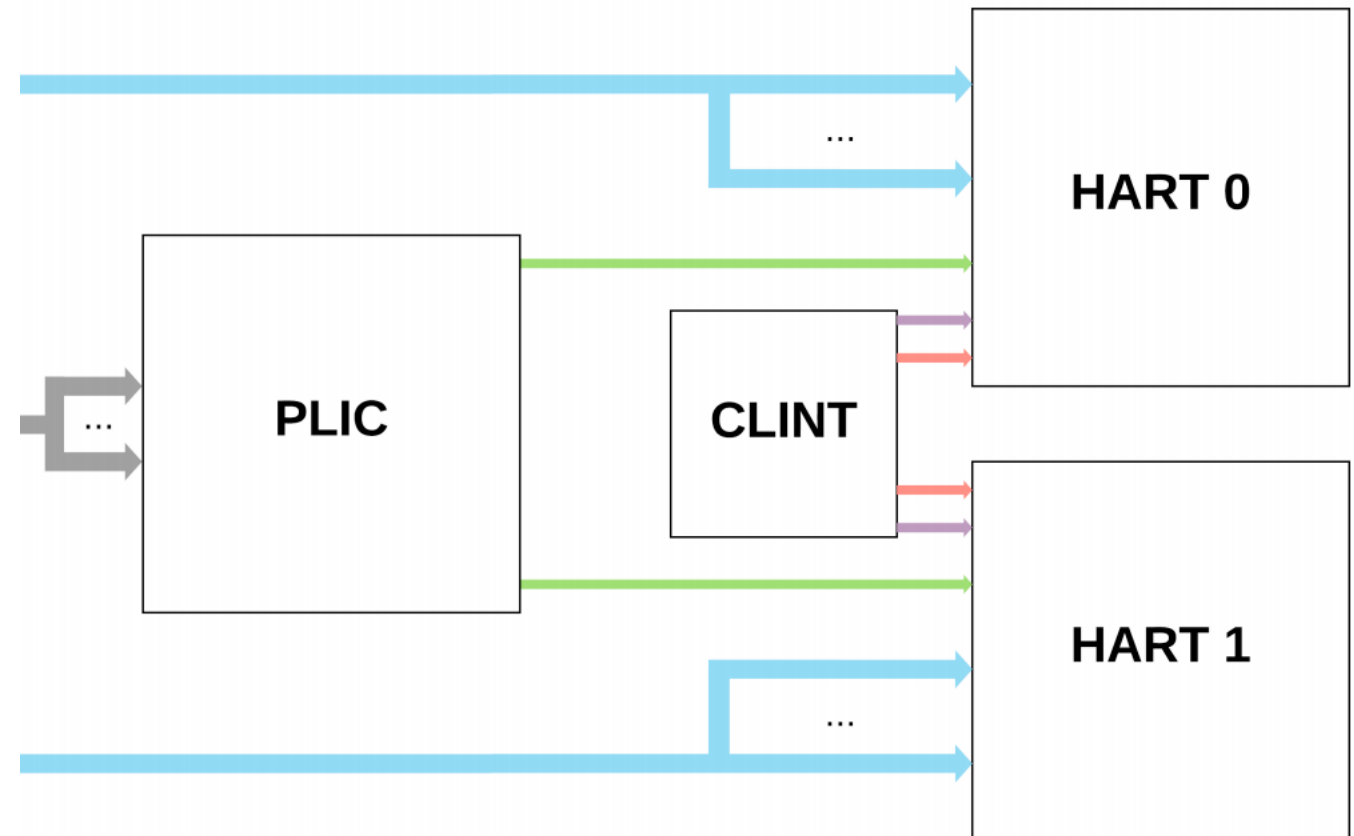
- Interrupts
  - Notify the CPU that something changed
  - Why interrupts?
    - Polling works, but not scalable or power efficient
    - Polling works if interrupts are very frequent
- Timer
  - A time base interrupt

# Interrupts “Generic” System



# RISC-V Interrupt/Timer

- CLINT (Core Local Interrupt)
- PLIC (Platform Interrupt)



# Clint handler

```
#define MCAUSE_INT_MASK 0x80000000 // [31]=1 interrupt, else exception
#define MCAUSE_CODE_MASK 0x7FFFFFFF // low bits show code

void software_handler() {
    unsigned long mcause_value = read_csr(mcause);
    if (mcause_value & MCAUSE_INT_MASK) { // INTERRUPT
        async_handler[(mcause_value & MCAUSE_CODE_MASK)]();
    } else { // EXCEPTION
        sync_handler[(mcause_value & MCAUSE_CODE_MASK)]();
    }
}
```

Interrupt	Code	Description
1	0	User software interrupt
1	1	Supervisor software interrupt
1	2	Reserved
1	3	Machine software interrupt
1	4	User timer interrupt
1	5	Supervisor timer interrupt
1	6	Reserved
1	7	Machine timer interrupt
1	8	User external interrupt
1	9	Supervisor external interrupt
1	10	Reserved
1	11	Machine external interrupt
1	>=12 && <16	Reserved
1	>=16	Implementation defined local interrupts
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	Load address misaligned
0	5	Load access fault
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode
0	9	Environment call from S-mode
0	10	Reserved
0	11	Environment call from M-mode
0	12	Instruction page fault
0	13	Load page fault
0	14	Reserved
0	15	Store/AMO page fault
0	>= 16	Reserved

# Let's do a ESP32 button

---

- Polling
- Interrupt



- Wifi