DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

IIC2343 - Arquitectura de Computadores (I/2026)

Ayudantía 11: Dispositivos I/O

Ayudantes: Alberto Maturana (alberto.maturana@uc.cl), Mario Rojas (mario.denzel@estudiante.uc.cl), Ignacio Gajardo (gajardo.ignacio@uc.cl)

Pregunta 1: Preguntas Conceptuales

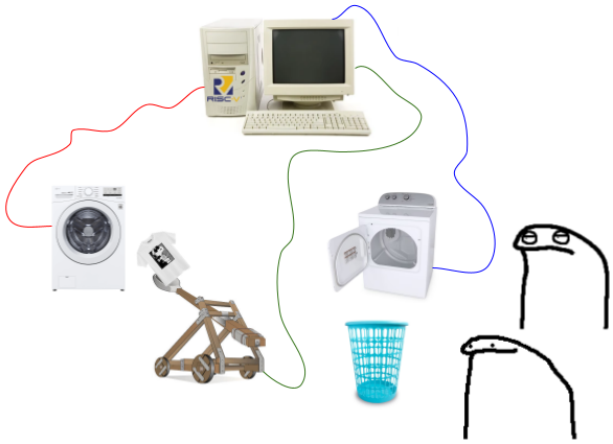
(a) ¿Por qué una CPU no puede determinar directamente qué dispositivo generó una interrupción únicamente a partir de la señal IRQ? ¿Qué mecanismo permite resolver este problema?

(b) ¿Por qué una *Interrupt Service Routine* (ISR) debe respaldar y restaurar el estado de la CPU al iniciar y antes de finalizar su ejecución, respectivamente?

(c) A diferencia del esquema *memory-mapped*, ¿por qué un dispositivo I/O con comunicación *port-mapped* necesita modificar la ISA con instrucciones adicionales?

Pregunta 2: Comunicación con dispositivos I/O (I2 2025-1)

Aburrido/a de los quehaceres del hogar, decide automatizar el lavado y secado de su ropa a través de un sistema centralizado que hace uso de la ISA RISC-V y de tres dispositivos: una lavadora (`washer`), una secadora (`dryer`) y una catapulta (`catapult`). A continuación, una ilustración de su innovadora idea:



Para llevar a cabo la comunicación entre la CPU y los dispositivos, los registros de 32 bits de estos últimos son mapeados en memoria desde la dirección 0xFEE1DEAD:

Offset	Nombre	Descripción
0x00	`washer_status`	Registro de estado de la lavadora
0x04	`washer_command`	Registro de comando de la lavadora
0x08	`washer_water_level`	Configuración de nivel de agua de la lavadora
0x0C	`dryer_status`	Registro de estado de la secadora
0x10	`dryer_command`	Registro de comando de la secadora
0x14	`dryer_type`	Configuración de tipo de ropa a secar
0x18	`catapult_status`	Registro de estado de la catapulta
0x1C	`catapult_command`	Registro de comando de la catapulta

Los estados, comandos y configuraciones, por otra parte, se rigen a partir de las siguientes tablas:

Nombre	Descripción	Valor
`washer_status`	Apagada	0
`washer_status`	Encendida, puerta abierta	1
`washer_status`	Encendida, puerta cerrada	2
`washer_status`	Lavando	4
`washer_command`	Encender	1
`washer_command`	Apagar	255
`washer_command`	Abrir puerta	16
`washer_command`	Cerrar puerta	32
`washer_command`	Expulsar ropa	64
`washer_command`	Lavar	128

Nombre	Descripción	Valor
`dryer_status`	Apagada	0
`dryer_status`	Encendida, puerta abierta	1
`dryer_status`	Encendida, puerta cerrada	2
`dryer_status`	Secando	4
`dryer_command`	Encender	1
`dryer_command`	Apagar	255
`dryer_command`	Abrir puerta	16
`dryer_command`	Cerrar puerta	32
`dryer_command`	Expulsar ropa	64
`dryer_command`	Secar	128

Nombre	Descripción	Valor
`catapult_status`	Preparada	11
`catapult_status`	Disparada	13
`catapult_command`	Lanzar	11
`catapult_command`	Reposicionar	13
`washer_water_level`	Alto	1
`washer_water_level`	Medio	2
`washer_water_level`	Bajo	3
`dryer_type`	Ropa normal	1
`dryer_type`	Sábanas	2
`dryer_type`	Toallas	3

Para llevar a cabo la automatización, desarrolla el siguiente programa que asume que la ropa ya se encuentra dentro de la lavadora y que el lanzamiento de la catapulta es instantáneo:

```
washing_automation:
# Carga 0 + 0xFEE1DEAD en t0
addi t0, zero, 0xFEE1DEAD
addi s0, zero, 0
addi s1, zero, 1
addi s2, zero, 2
addi s3, zero, 3
addi s4, zero, 4
addi s5, zero, 16
addi s6, zero, 32
addi s7, zero, 64
addi s8, zero, 128
addi s9, zero, 255
addi s10, zero, 11
addi s11, zero, 13

step_one:
    lw t1, 0(t0) # Carga Mem[t0 + 0] en t1
    # Salta a continue_step_one si t1 >= s1
    bge t1, s1, continue_step_one
    sw s1, 4(t0) # Carga s1 en Mem[t0 + 4]
    continue_step_one:
      # Salta a step_two si t1 == s2
      beq t1, s2, step_two
      sw s6, 4(t0)

step_two:
    lw t1, 24(t0)
    beq t1, s10, step_three
    sw s11, 28(t0)

# Continúa en step_three -->
```

```
step_three:
    lw t1, 12(t0)
    bge t1, s1, continue_step_three
    sw s1, 16(t0)
    continue_step_three:
      beq t1, s1, step_four
      sw s5, 16(t0)

step_four:
    sw s1, 8(t0)
    sw s8, 4(t0)
    while_step_four:
        lw t1, 0(t0)
        beq t1, s4, while_step_four
    sw s5, 4(t0)
    sw s7, 4(t0)
    sw s9, 4(t0)

step_five:
    sw s10, 28(t0)
    sw s11, 28(t0)

step_six:
    sw s6, 16(t0)
    sw s3, 20(t0)
    sw s8, 16(t0)
    while_step_six:
        lw t1, 12(t0)
        beq t1, s4, while_step_six
    sw s5, 16(t0)
    sw s7, 16(t0)
    sw s9, 16(t0)
```

Describa, basándose en las tablas provistas, las tareas que realiza el programa en cada *label* `step_*`. Haga una descripción según los estados, comandos y configuraciones de los dispositivos; no una explicación textual de cada instrucción. Sea preciso/a respecto a las condiciones evaluadas en los saltos e indique, cuando corresponda, el momento en el que se continúa al siguiente `step`.

Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.

