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

IIC2343 - Arquitectura de Computadores (I/2025)

Interrogación 2

Pauta de evaluación

Pregunta 1: Arquitecturas de Computadores (6 ptos.)

(a) (1.5 ptos.) Un programa, desarrollado en un lenguaje de programación específico, es compilado en dos computadores distintos: uno con ISA RISC, y otro con ISA CISC. ¿Cuál de las dos compilaciones resultará en un código de máquina de mayor tamaño, *i.e.* con mayor cantidad de instrucciones? Justifique su respuesta.

Solución: El programa para la ISA RISC tendrá un tamaño de código de máquina mayor, ya que RISC define *sets* de instrucciones muy simples que redundan en una mayor cantidad de instrucciones en comparación a CISC para hacer una tarea específica (por ejemplo, multiplicaciones con sumas sucesivas en contraste con una instrucción directa que realice dicho cómputo). Se otorgan **0.75 ptos.** por responder correctamente y **0.75 ptos.** por justificación.

(b) (1.5 ptos.) Suponga que se modifica el computador básico estudiado en clases para que funcione con una única unidad de memoria, *i.e.* que su microarquitectura siga el modelo Von Neumann. ¿Implicaría esto cambios en su ISA? Justifique.

Solución: No necesariamente. La ISA especifica la arquitectura del *set* de instrucciones que soporta el computador, por lo que la microarquitectura debe adaptarse a ella para su ejecución. El cambio del paradigma Harvard a Von Neumann implica que la microarquitectura debe modificarse para seguir soportando la ISA, y no al revés. Se otorgan **0.75 ptos.** por responder correctamente, y **0.75 ptos.** por justificación.

(c) (1.5 ptos.) Suponga que se le solicita revisar la arquitectura del computador básico estudiado en clases y, en caso de ser necesario, modificar su ISA para **reducir su consumo energético**, puesto que busca ser adaptada para dispositivos móviles. Indique si es necesario realizar cambios y de qué tipo; en caso contrario, explique por qué.

Solución: No es necesario realizar cambios. El motivo es que la ISA de tipo RISC ya de por sí es más óptima en términos de consumo energético respecto a CISC, dado que esta última soporta instrucciones más complejas con más ciclos de ejecución, implicando un mayor uso de energía por instrucción. Se otorgan **0.75 ptos.** por responder correctamente, y **0.75 ptos.** por justificación.

(d) (1.5 ptos.) Una arquitectura de tipo *Register-Memory* permite usar datos de la memoria como operandos de forma directa. En una arquitectura de tipo *Register-Register*, en cambio, es necesario que los datos se almacenen **antes** en los registros para poder operar con ellos. Indique, a partir de esta categorización y de forma argumentada, el tipo de arquitectura que presenta el computador básico estudiado en clases.

Solución: La arquitectura del computador básico es de tipo ***Register-Memory*** dado que la conexión entre la salida de la memoria de datos se conecta directamente con el multiplexor del operando `B` de la ALU. Esto permite la ejecución de instrucciones como `ADD A,(Dir)`, donde se usan valores de memoria como operandos de la ALU en un solo ciclo. Se otorgan **0.75 ptos.** por responder correctamente, y **0.75 ptos.** por justificación.

Pregunta 2: Comunicación con dispositivos I/O - Parte 1 (6 ptos.)

(a) (1.5 ptos.) En el pasado, la comunicación con dispositivos I/O se diseñó mediante mapeo a puertos como alternativa al mapeo a memoria, debido a una desventaja que este último presentaba en ese entonces. Hoy en día, esa limitación ha dejado de ser relevante, por lo que el mapeo a memoria es el que predomina en arquitecturas modernas. Explique cuál era la desventaja que tenía el mapeo a memoria en ese contexto, y por qué en la actualidad ya no representa un problema.

Solución: La desventaja que tenía el mapeo a memoria consistía en la *memory barrier*, *i.e.* la limitación del espacio direccionable por la reserva significativa de direcciones. Como en el pasado las memorias RAM tenían un tamaño reducido, esta pérdida era significativa e implicaba la imposibilidad de usar variables normalmente por la baja cantidad de direcciones disponibles. Hoy en día, en cambio, las memorias RAM poseen un tamaño lo suficientemente grande para que este problema sea inexistente. Se otorgan **0.75 ptos.** por responder correctamente, y **0.75 ptos.** por justificación.

(b) (1.5 ptos.) Explique el uso e importancia de la señal IRQ enviada por el controlador de interrupciones a la CPU.

Solución: La señal IRQ (*Interrupt Request*) es utilizada para comunicarle a la CPU que existe, al menos, un dispositivo I/O que requiere su atención. El controlador de interrupciones se encarga de su envío en caso de uno de los dispositivos de la arquitectura haya gatillado una interrupción. Es importante ya que es la que permite que inicie el proceso de interrupción, que termina en la ejecución de la ISR (*Interrupt Service Routine*) del dispositivo. Se otorgan **0.75 ptos.** por responder correctamente, y **0.75 ptos.** por justificación.

(c) (1.5 ptos.) Entregue dos ejemplos de interrupciones gatilladas por la CPU y no por dispositivos I/O, indicando su tipo (excepción o *trap*) y explicando en qué consisten. Ambos ejemplos pueden ser de un mismo tipo.

Solución: A continuación, se listan dos ejemplos:

- ***Stack overflow*:** Es una **excepción** que se gatilla cuando el tamaño de la memoria *stack* supera el límite permitido, generando posibles errores de sobreescritura en variables.
- ***Environment call*:** Es una ***trap*** que permite cederle el control al sistema operativo para que realice una acción, como imprimir en consola o terminar un proceso.

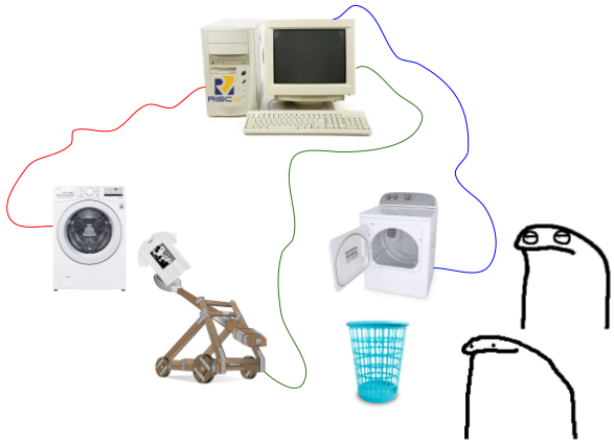
Se otorgan **0.75 ptos.** por cada ejemplo correctamente descrito. Se otorga la mitad del puntaje en un ejemplo si su descripción no es correcta, o bien su tipo no es consistente con el descrito.

(d) (1.5 ptos.) Suponga que, al computador básico estudiado en clases, se le añade todo el soporte en *hardware* necesario para comunicarse con dispositivos I/O mapeados en memoria a través de interrupciones. Además, uno de los dispositivos mapeados corresponde a un *pendrive*, del cual se desea copiar información a la RAM. ¿Es estrictamente necesario incluir *hardware* adicional para cumplir con esta funcionalidad? Justifique su respuesta.

Solución: En estricto rigor no es necesario. Si bien lo óptimo es contar con un controlador de DMA (*Direct Memory Access*) para orquestar la transferencia entre dispositivos y la memoria RAM, en la práctica se puede hacer uso de *Programmed* I/O para llevarla a cabo, *i.e.* usar los registros de CPU como puente para transferir los datos. Si bien esto en la práctica no se realiza por la cantidad de ciclos de CPU perdidos, se puede implementar como alternativa en caso de querer un *hardware* más sencillo. Se otorgan **0.75 ptos.** por responder correctamente, y **0.75 ptos.** por justificación.

Pregunta 3: Comunicación con dispositivos I/O - Parte 2 (6 ptos.)

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`.

Solución:

- `step_one` **(1 pto.)**: Se revisa el estado de la lavadora. Si está apagada, se prende. Luego, si está prendida con la puerta abierta, se cierra y salta a `step_two`. En otro caso, continúa a `step_two` directamente.
- `step_two` **(0.5 ptos.)**: Se revisa el estado de la catapulta. Si está preparada, se salta a `step_three`. En otro caso, se reposiciona y continúa a `step_three`.
- `step_three` **(1 pto.)**: Se revisa el estado de la secadora. Si está apagada, se prende. Luego, si está prendida con la puerta cerrada, se abre y salta a `step_four`. En otro caso, continúa a `step_four` directamente.
- `step_four` **(1.5 ptos.)**: Se configura un nivel alto de agua en la lavadora e inicia el lavado. Una vez que termina se abre la puerta, se expulsa la ropa (para que quede en la catapulta) y se apaga.
- `step_five` **(0.5 ptos.)**: Se lanza el contenido de la catapulta y se reposiciona.
- `step_six` **(1.5 ptos.)**: Se cierra la puerta de la secadora, se configura para toallas e inicia el secado. Una vez que termina se abre la puerta, se expulsa la ropa (para que quede en el canasto) y se apaga.

Para los pasos 1, 3, 4 y 6, se descuenta un 25 % del puntaje si existe como máximo un error o detalle faltante; un 50 % si existen como máximo dos errores; y un 100 % en otro caso. Para los pasos 2 y 5, se descuenta un 50 % del puntaje si existe como máximo un error o detalle faltante; y un 100 % en otro caso.