



IIC2343 - Arquitectura de Computadores (I/2023)

**Examen**

Respuestas sin desarrollo o justificación no tendrán puntaje.

**Instrucciones**

Lea atentamente el enunciado. Responda cada pregunta en hojas separadas. Debe poner su nombre, número de alumno y sección en cada hoja. Cada pregunta tiene 10 puntos; pero solo se corregirá hasta el máximo que está indicado en el título de la pregunta. Estos máximos entre las 7 preguntas suman 48 puntos, que corresponden a la nota 7; el 3,7 se alcanza con 21,6 puntos. Usted puede decidir qué letras de las preguntas contestar para alcanzar esta cantidad de puntos.

**Pregunta 1: Preguntas conceptuales (7 ptos.)**

- (a) (1 pto.) En el videojuego *The Legend of Zelda*<sup>TM</sup> de la consola *Nintendo Entertainment System*<sup>TM</sup>, se hace uso de rupias como moneda de cambio. El contador de rupias siempre es mayor o igual a cero, pero cuenta con la particularidad de que deja de incrementarse una vez que llega a 255. Explique, a partir de los contenidos del curso, por qué podría suceder esto.
- (b) (2 ptos.) Indique si los siguientes números pueden representarse o no como un número `float` bajo el estándar IEEE754 con su valor exacto: (i)  $2^{26} + 2$ ; (ii)  $2^{70}$ . Si se puede representar, indique la representación; en otro caso, justifique por qué no se puede.
- (c) (1 pto.) Suponga que le entregan un computador construido por estudiantes del curso “Arquitectura de Computadores”. Decide utilizarlo y define una variable de 4 bytes igual a 1024. No obstante, al imprimir su valor, la consola le arroja el valor 262144 (equivalente a  $2^{18}$ ). Explique qué podría estar mal con la implementación de los/as estudiantes y cómo podría resolverse.
- (d) (2 ptos.) Indique, justificadamente, qué señales y componentes son esenciales en la comunicación entre los dispositivos I/O y la CPU para que esta pueda atender interrupciones ejecutando ISRs.
- (e) (1 pto.) Suponga que se implementa memoria virtual en el computador básico del curso. ¿Las direcciones de los registros SP y PC deben ser virtuales o físicas en esta implementación? Justifique su respuesta.

- (f) (2 ptos.) Señale qué modificaciones mínimas, a nivel de *hardware*, se deben realizar sobre el computador básico con *pipeline* para que este sea *2-issue*. Puede asumir que el compilador se hace cargo del envío de *bundles* de 2 instrucciones como VLIW.
- (g) (1 pto.) Explique brevemente por qué las GPU suelen categorizarse como arquitecturas SIMT dentro del paradigma SIMD.

## Pregunta 2: Circuitos eléctricos y almacenamiento (5 ptos.)

- (a) (5 ptos.) Diseñe, utilizando desde compuertas lógicas **hasta flip-flops**, un contador secuencial de 2 bits **inverso**, que parte con valor 11 y que decrementa con cada flanco de subida de la señal de control *clock* hasta llegar a 00 y luego hacer *overflow* para partir de nuevo, *i.e.* 11-10-01-00-11-10-01-00-etc.
- (b) (5 ptos.) Se desea tener un registro de estados de 4 bits de tipo *one-hot*, *i.e.* que siempre tiene un **único** bit igual a 1, no puede existir más de uno ni contener solo ceros. Modifique el diagrama de un registro para que este habilite su señal de carga L si, y solo si el valor de entrada cumple con ser *one-hot*.

## Pregunta 3: Computador básico (7 ptos.)

- (a) (3 ptos.) Modifique la arquitectura del computador básico para implementar las instrucciones NOT B,B, SHL B,B y SHR B,B, es decir, que realice las operaciones de un operando sobre el registro B y almacene el resultado en él. Modifique el diagrama adjunto o señale todas las conexiones y componentes que se deben añadir. Incluya la combinación de señales que las ejecutan.
- (b) (3 ptos.) Modifique la arquitectura del computador básico para implementar las instrucciones MOV A, (B+offset) y MOV (B+offset), A, *i.e.* instrucciones de direccionamiento indirecto con registro B y *offset*, siendo este último un literal que puede ser positivo o negativo. Modifique el diagrama adjunto o señale todas las conexiones y componentes que se deben añadir. Incluya la combinación de señales que las ejecutan.

- (c) (4 ptos.) Asuma que se agregan las instrucciones anteriores a la ISA del computador básico y se ejecuta el programa adjunto. Señale solo los valores finales de los registros A, B y SP. Asuma que todos los registros son de 8 bits y que SP se interpreta como un número positivo.

```
MOV A,2          // Dirección Mem. Instr.: 0x00
MOV B,5          // Dirección Mem. Instr.: 0x01
PUSH A           // Dirección Mem. Instr.: 0x02
PUSH B           // Dirección Mem. Instr.: 0x03
CALL mystery     // Dirección Mem. Instr.: 0x04
POP B            // Dirección Mem. Instr.: 0x05-0x06
POP A            // Dirección Mem. Instr.: 0x07-0x08
JMP end          // Dirección Mem. Instr.: 0x09
mystery:
  MOV B,0        // Dirección Mem. Instr.: 0x0A
  NOT B,B        // Dirección Mem. Instr.: 0x0B
  MOV A,(B + -2) // Dirección Mem. Instr.: 0x0C
  ADD A,4        // Dirección Mem. Instr.: 0x0D
  MOV (B + -2),A // Dirección Mem. Instr.: 0x0E
  RET           // Dirección Mem. Instr.: 0x0F-0x10
end:
```

#### Pregunta 4: Memoria caché y memoria virtual (8 ptos.)

- (a) (4 ptos.) Suponga que posee una memoria caché *2-way associative* de 8 líneas y 4 palabras por línea para una memoria principal de 64 bytes. Indique el *hit-rate* y el estado final de la caché (bits de *tag* y bit de validez) para los siguientes accesos de memoria: 0, 49, 19, 34, 2, 51, 16, 35. Asuma que se implementa una política de reemplazo LRU.
- (b) (2 ptos.) Indique si se puede aumentar el *hit-rate* de la pregunta anterior solo cambiando la función de correspondencia. No es necesario que lo calcule, pero sí que justifique su respuesta.
- (c) (4 ptos.) Suponga que se tiene un espacio de direcciones virtuales de 30 bits y direcciones físicas de 20 bits. Indique, en bytes, el tamaño de página **mínimo** que permite que la tabla de páginas de un proceso “quepa” por completo en una página, *i.e.* que su tamaño no supere el tamaño de la tabla. Asuma que en cada entrada de la tabla de páginas se utilizan 3 bits de *metadata*.

## Pregunta 5: RISC-V e I/O (6 ptos.)

- (a) (2 ptos.) Indique justificadamente si el siguiente fragmento de código respeta la convención de llamada de RISC-V.

```
.data
N:      .word 23
divisor: .word 12
.text
main:
    addi sp, sp, -4
    sw ra, 0(sp)
    call remainder
    lw ra, 0(sp)
    addi sp, sp, 4
    j end
remainder:
    lw a0, N
    lw a1, divisor
    rem a0, a0, a1
    ret
end:
```

- (b) (2 ptos.) Indique si el siguiente fragmento de código en RISC-V termina su ejecución. Si lo hace, indique el valor del registro a0. En otro caso, justifique por qué no termina.

```
.data
N: .word 4
.text
main:
    addi sp, sp, -4
    sw ra, 0(sp)
    lw t0, N
    mv a0, t0
    call factorial
    lw ra, 0(sp)
    addi sp, sp, 4
    j end
factorial:
    addi sp, sp, -4
    sw a0, 0(sp)
    blez a0, factorial_zero
    addi a0, a0, -1
    call factorial
    lw t0, 0(sp)
    mul a0, a0, t0
    j factorial_end
factorial_zero:
    li a0, 1
factorial_end:
    addi sp, sp, 4
    ret
end:
```

Para las siguientes preguntas, suponga que tiene una impresora cuyos registros de estado y comando de 32 bits están mapeados en memoria desde la dirección 0x10150000. A continuación, se indican las tablas de direcciones, comandos y estado:

Offset	Nombre	Descripción
0x00	printer_stat	Registro de estado.
0x04	printer_comm	Registro de comando.
0x08	printer_clr	Registro de color.
0x0C	printer_sz	Registro de tamaño hoja.
0x10	printer_or	Registro de orientación hoja.

Nombre	Descripción	Valor
printer_stat	Apagada.	0
printer_stat	Prendida.	1
printer_stat	Prendiendo.	127
printer_stat	Imprimiendo.	255
printer_comm	Prender.	0
printer_comm	Apagar.	1
printer_comm	Imprimir.	2
printer_clr	Blanco y negro.	0
printer_clr	Color.	1
printer_sz	Tamaño carta.	0
printer_sz	Tamaño oficio.	1
printer_or	Horizontal.	0
printer_or	Vertical.	1

Suponga que bajo este contexto se ejecuta el siguiente programa en RISC-V:

```
.text
main:
    li t0, 0x10150000
    lw t1, 0(t0)
    li s0, 1
    li s1, 127
    li s2, 0
    beq t1, s0, actionThree
    beq t1, s1, actionTwo
    beq t1, s2, actionOne
    j end
actionOne:
    li t2, 0
    sw t2, 4(t0)
actionTwo:
    lw t1, 0(t0)
    beq t1, s0, actionThree
    j actionTwo
actionThree:
    li s4, 1
    sw s4, 8(t0)
    li s5, 0
    sw s5, 12(t0)
    li s6, 1
    sw s6, 16(t0)
    li t2, 2
    sw t2, 4(t0)
end:
```

Respecto a este programa:

- (c) (2 ptos.) Indique justificadamente qué tipo de comunicación se presenta entre la CPU y la impresora en el código anterior.
- (d) (4 ptos.) Explique en términos prácticos, según el programa y las tablas adjuntas, lo que realizan las acciones `actionOne`, `actionTwo` y `actionThree`.

## Pregunta 6: Paralelismo y Coherencia de caché (8 ptos.)

- (a) (2 ptos.) Describa cómo se puede modificar el computador básico con *pipeline* para que las predicciones de salto erróneas pierdan dos ciclos en vez de tres.
- (b) (4 ptos.) A partir del siguiente programa del computador básico con *pipeline*:

```
DATA:
i    0
arr  13
    13
len  1
CODE:
main:
    MOV A,(i)
    MOV B,(len)
    JEQ A,B,end // if A == B go to end
    MOV B,arr
    ADD B,A
    ADD A,1
    MOV (i),A
    MOV A,(B)
    SUB A,2
    MOV (B),A
    JMP main
end:
    MOV A,0
    MOV B,0
    ADD (i)
```

Determine el número de ciclos que demora su ejecución detallando en un diagrama los estados del *pipeline* por instrucción. Asuma que el manejo de *stalling* es por *software* a través de la instrucción NOP y que la unidad predictora de saltos asume que estos nunca se realizan. Indique en el diagrama cuando ocurre *forwarding* (con flechas del registro a la etapa que correspondan), *stalling* y *flushing* (con tachado en las instrucciones *flushheadas*).

- (c) (4 ptos.) Suponga que posee una arquitectura MIMD de memoria compartida con 2 CPU que poseen su propia caché y que siguen el protocolo MESI para asegurar consistencia. La memoria contiene los siguientes datos de variables y las CPU0 y CPU1 ejecutan los siguientes programas:

Dirección	Label	Valor
0x00	i	1
0x01	arr	1
0x02		10

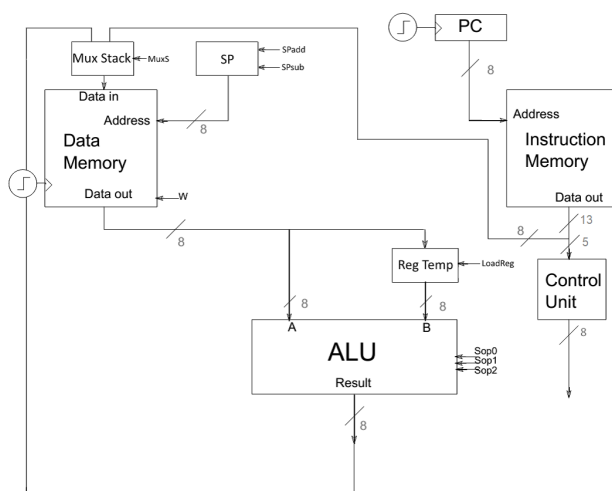
```
// CPU0    // CPU1
MOV B,(i)   MOV A,(arr)
MOV A,(arr) MOV B,arr
ADD B,A     ADD B,A
SHR A,A     MOV (B),A
MOV (B),A   NOP
```

Asumiendo que cada dirección se almacena en una línea distinta, indique el estado de cada línea de las caché (M/E/S/I) para cada iteración completando la tabla adjunta, asumiendo que todas las líneas parten en estado I:

Instrucción	CPU0			CPU1		
	i	arr[0]	arr[1]	i	arr[0]	arr[1]
1						
2						
3						
4						
5						

## Pregunta 7: Déjà vu (7 ptos.)

- (a) (3 ptos.) Dados números naturales  $K, N, T$ , donde  $K \geq 2$  y  $N > T > 0$ , describa un algoritmo para transformar de manera **eficiente** números naturales codificados en base  $K^N$  a  $K^T$ .
- (b) (4 ptos.) Una máquina de *stack* es un computador que utiliza una memoria de *stack* en vez de registros para almacenar los resultados de las operaciones. Suponga que le entregan, **sin que el equipo docente se entere**, el siguiente diagrama e ISA:



Instrucción	Opcod	Señales de control
PUSH Lit	00000	SPsub = 1
	00001	W = 1, MuxS = 1
POP	00010	SPadd = 1
ADD	00011	SPadd = 1, LoadReg = 1
	00100	Sop = ADD, W = 1, MuxS = 0
SUB	00101	SPadd = 1, LoadReg = 1
	00110	Sop = SUB, W = 1, MuxS = 0
AND	00111	SPadd = 1, LoadReg = 1
	01000	Sop = AND, W = 1, MuxS = 0
OR	01001	SPadd = 1, LoadReg = 1
	01010	Sop = OR, W = 1, MuxS = 0
XOR	01011	SPadd = 1, LoadReg = 1
	01100	Sop = XOR, W = 1, MuxS = 0
SHL	01101	Sop = SHL, W = 1, MuxS = 0
SHR	01110	Sop = SHR, W = 1, MuxS = 0
NOT	01111	Sop = NOT, W = 1, MuxS = 0
NOP	10000	W = 0

De lo anterior se destaca que las instrucciones de tipo PUSH toman dos ciclos, mientras que las del computador básico toman un ciclo. Explique qué diferencias prácticas existen en el manejo de SP del computador básico y el de la máquina presentada, profundizando en cómo cambia la forma en la que se maneja la memoria de *stack*.

- (c) (3 ptos.) Considere un sistema con páginas de 16KB, direcciones virtuales de 48 bits, direcciones físicas de 30 bits y 4 bits de *metadata* para las traducciones. Proponga un esquema de paginación **multinivel** que asegure que la tabla de páginas de cada nivel “quepa” en una página, *i.e.* que su tamaño no supere los 16KB.

## Torpedo - Diagramas de utilidad

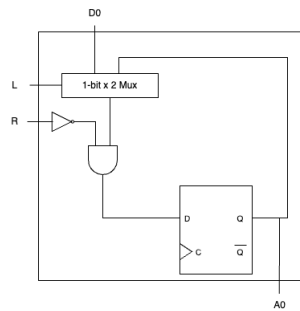


Figura 1: Diagrama interno de un registro, se puede extender para 4 bits.

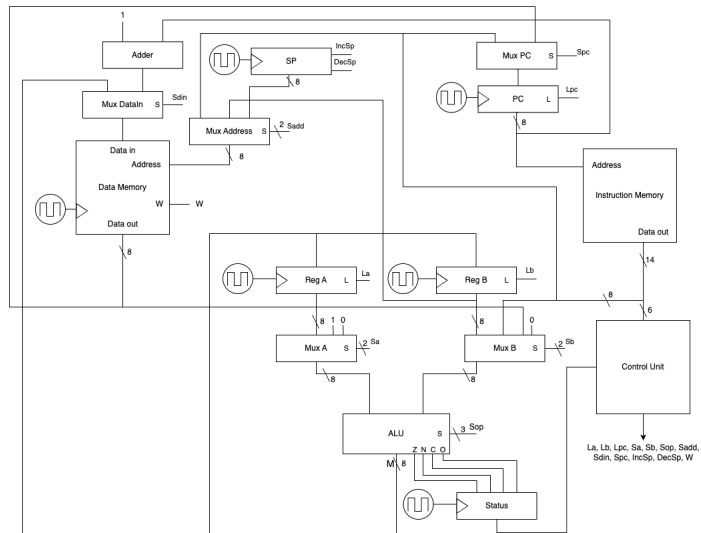


Figura 2: Diagrama del computador básico con saltos y subrutinas.

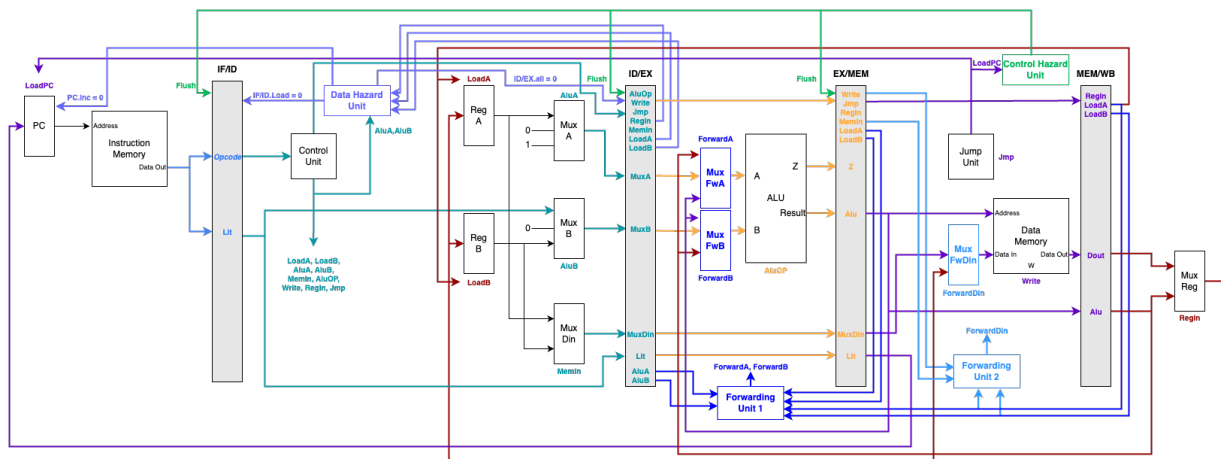


Figura 3: Diagrama del computador básico con *pipeline*.