



IIC2343-2 - Arquitectura de Computadores (I/2022)

Interrogación 1

Respuestas sin desarrollo o justificación no tendrán puntaje.

Miércoles 13 de Abril a las 18:30 horas

Instrucciones

Lea atentamente los enunciados. Responda cada pregunta en hojas separadas. Ponga su nombre, número de alumno y número de lista. Siga el código de honor.

Código de Honor de la UC

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

Pregunta	Puntos	Logrados
Ups... se me rompieron las teclas + y -	10	
Bienvenido a la NASA	16	
¿Pair programming?	16	
B#	18	
Total:	60	

Nombre: _____ N° de alumno: _____ N° lista: _____

Pregunta 1: Ups... se me rompieron las teclas + y - (10 ptos.)

Utilizando únicamente los operadores lógicos *bitwise* escriba dos funciones en Python:

(a) Una función que permita sumar dos números.

(6)

Solución:

- Entender y aplicar correctamente la suma (2 ptos.)
- Entender y utilizar correctamente el carry (2 ptos.)
- Entender y recorrer correctamente los bits (2 ptos.)

(b) Una función que permita restar dos números. Puede llamar a la función anterior.

(4)

Solución:

- Entender y aplicar complemento 2 (2 ptos)
- Usar correctamente los operadores (2 ptos)

* Entender se refiere a que explica lo que quiere hacer, de esta forma si tenía una buena idea pero no pudo implementarla, quedaría con 1 punto en dicho ítem

Pregunta 2: Bienvenido a la NASA (16 ptos.)

Usando tan solo compuertas lógicas **NAND**, implemente:

(a) Un sumador de 4 bits. Puede implementar componentes intermedios.

(12)

Solución:

- Funcionamiento del Full-Adder usando NANDS (6 ptos).
 - Compuerta XOR (2 ptos).
 - HA y resto de las compuertas (1 pto cada una).
- Estructura del Full-Adder (3 ptos). Se considera la notación, los componentes y la lógica detrás.
- Estructura del sumador (3ptos). Idem.

(b) Un restador de 4 bits. Puede usar el componente anterior y/o los que haya definido.

(4)

Solución:

- Se suma 1, Carry-in = 1 (1 pto).
- Se invierte A o B (1 pto).
- Componentes nuevos se implementan con NANDS (1 pto).
- Los operadores lógicos están conectados correctamente, con una intención definida, es decir, no existen conexiones de componentes lógicos de forma aleatoria (1 pto).

Pregunta 3: ¿Pair programming? (16 ptos.)

Escriba una subrutina en el **Assembly** del computador básico que sea capaz de contar la cantidad de números pares en un arreglo. Al momento de llamar la subrutina, el largo del arreglo estará en el registro A y la ubicación en memoria del primer elemento del arreglo estará en el registro B. Al finalizar la subrutina, el resultado debe estar en el registro A. Recuerde que la subrutina a escribir se debe poder usar más de una vez.

Solución:

- Definición de subrutina haciendo uso de un label (1 pto)
- Uso de la instrucción RET para terminar la subrutina (1 pto)
- Uso correcto de puntero del arreglo (1 pto)
- Uso correcto de argumentos (2 ptos)
- Uso correcto de ciclos iteradores: los ciclos terminan en el largo de iteraciones es igual al largo del arreglo y uso correcto de saltos (3 ptos)
- Se hace la decisión de paridad correctamente (2 ptos)
- El resultado queda en el registro A (2 ptos)
- La subrutina es reutilizable (2 ptos)
- Correctitud del código (2 ptos)

Pregunta 4: B# (18 ptos.)

Se requiere agregar las siguientes instrucciones al computador básico. Para cada caso realice los cambios en el diagrama adjunto, especifique las nuevas señales de control agregadas (si corresponde) e indique en una tabla las señales de control y el estado que deben tomar para ejecutar la instrucción.

- (a) Agregue la instrucción **JMP B**, la cual salta a la dirección almacenada en el registro B.

(6)

Solución:

Señales

- Define que la señal del selector PC debe ser el registro B (1 pto)
- Define que no modifica registros ni memoria (1 pto)
- Define que la señal de carga del PC debe estar activa (1 pto)

Modificación:

- Es realizable (1 pto)
- Una salida del registro B va hacia una expansión del selector PC o se crea un selector adicional conectado al PC (2 ptos)

- (b) Agregue la instrucción **CALL B**, la cual difiere de **CALL Dir** en que la dirección de destino se obtiene desde el registro B.

(6)

Solución:

Señales

- Define que la señal del selector PC debe ser el registro B (1 pto)
- Define que no modifica registros (1 pto)
- Define correctamente las señales propias de una instrucción **CALL** (1 pto)

Modificación:

- Es realizable (1 pto)
- Una salida del registro B va hacia una expansión del selector PC o se crea un selector adicional conectado al PC (2 ptos)

- (c) Agregue la instrucción **INC (B)**, la cual incrementa en 1 el valor en memoria en la posición indicada por el registro B, se debe ejecutar en 1 ciclo.

(6)

Solución:

Opción 1:

Señales

- No carga registros (1 pto)
- Selector de DataIn es correcto (1 pto)
- Selector de A y Selector de B correctos (1 pto)
- Carga en memoria (1 pto)

Explicación

Identifica que no es necesario modificar hardware, justificadamente (2 ptos)

Opción 2:

En caso de hacer modificación de hardware se da 2 ptos por la modificación y 2 ptos por uso correcto de señales (no se obtiene puntaje completo).