



IIC2343 - Arquitectura de Computadores (II/2022)

Interrogación 1

Respuestas sin desarrollo o justificación no tendrán puntaje.

Martes 20 de Septiembre 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. Existen 4 preguntas de las cuales deberá contestar un máximo de 3 preguntas.

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
Estándar IEEE-{inserte No de alumno}	20	
Distancia de Hamming	20	
Circuito Combinacional	20	
B# 2.0	20	
Total:	80	

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

Pregunta 1: Estándar IEEE-{inserte No de alumno} (20 ptos.)

En clases se vio el estándar IEEE-754 para representar números de punto flotante de 32 bits. Defina su propio estándar para representar números complejos en 32 bits.

Recordatorio: Un número complejo tiene la siguiente estructura:

$$x = a + bi \quad \text{donde } a, b \in \mathbb{R}$$

- (a) Describa todos los elementos que debe tener su representación. Defina la cantidad de bits y la codificación de cada uno de estos. (6)

Solución:

Una solución posible es:

- 1 bit para el signo de la parte real del número.
- 5 bits para el exponente de la parte real del número. Este está desplazado en 15.
- 10 bits para la mantisa de la parte real del número, se considera bit implícito.
- 1 bit para el signo de la parte imaginaria del número.
- 5 bits para el exponente de la parte imaginaria del número. Este está desplazado en 15.
- 10 bits para la mantisa de la parte imaginaria del número, se considera bit implícito.

Para ambos se reserva los exponentes 0 y mantisa 0 para representar el cero.

- (b) Comente cuál es el rango y la precisión de su representación. (7)

Solución:

- Rango: El mayor número que se puede representar es el 131008 y el menor número positivo es el 0.000000059604645 (aproximadamente).
- Precisión: Esta representación tiene 11 bits de precisión.

- (c) Escriba la palabra de 32 bits que en su representación codifique el número $513,0 - 4,8750i$. (7)

Solución:

Podemos representar el 512_{10} como $1000000001_2 = 1,000000001_2 \times 2^9$.

A la vez, $4,8750_{10}$ en binario es $100,1110_2 = 1,001110 \times 2^2$.

Desplazamos los exponentes, $9 + 15 = 24$ y $2 + 15 = 17$.

Entonces:

- 0: signo de la parte real.
- 11000: exponente de la parte real.
- 0000000010: mantisa de la parte real.
- 1: signo de la parte imaginaria.
- 10001: exponente de la parte imaginaria.
- 0011100000: mantisa de la parte imaginaria.

El numero se vería como:

01100000000000101100010011100000

Pregunta 2: Distancia de Hamming (20 ptos.)

La distancia de Hamming entre dos palabras del mismo largo, corresponde a la cantidad de bits en las mismas posiciones que son distintos. Por ejemplo para las palabras 00101000 y 01010100 la distancia de Hamming es 5, ya que los bits en las posiciones 2, 3, 4, 5 y 6 difieren entre ellos.

Escriba una subrutina en el **Assembly** del computador básico que sea capaz de calcular la distancia de Hamming entre dos números. Al momento de llamar la subrutina, el primer número se encontrará registro A y el segundo 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.

Escriba también un pequeño programa que llame a esta subrutina con los parámetros que Ud. elija.

Solución:

```
DATA:
    distancia 0
CODE:

hamming:
    XOR B,A          // Calculamos los bits distintos
    MOV A,0          // |
    MOV (distancia), A // | iniciamos la distancia en 0
loop:
    MOV A,B          // |
    AND A,1          // |
    CMP A,0          // |
    JEQ par          // | si la diferencia es impar
    MOV A,(distancia) // |
    ADD A,1          // |
    MOV (distancia),A // | distancia += 1
par:
    MOV A,B          // |
    SHR A,A          // |
    MOV B,A          // | diferencia = diferencia//2
    CMP A,0          // |
    JEQ loop_end     // |
    JMP loop          // | si la diferencia es != 0 repetir lo anterior
loop_end:
    MOV A, (distancia) // retornamos la distancia
    RET

MOV A,255
MOV B,15
CALL hamming
end:
JMP end
```

Pregunta 3: Circuito Combinacional (20 ptos.)

En clases se vio una forma de representar números binarios en base 2, en el que el número a representar es el antecesor más 1, pero esta no es la única forma de representar un número en binario. Una alternativa es la representación Gray Code, en que los dígitos se organizan de tal manera que al incrementar en 1, sólo un bit de la representación cambia de 1 a 0 o de 0 a 1 (distancia de Hamming 1). La siguiente tabla muestra la equivalencia entre representación en base 2 y Gray Code:

Código Gray				Base 2			
X_3	X_2	X_1	X_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

De acuerdo a la tabla diseñe un circuito que dado un input X de 4 bits expresado en código Gray, el output Y de 4 bits sea el número en base 2:

Alimento para la mente: La representación de Gray es muy útil cuando se quiere verificar si algo ha cambiado de estado, por lo que es común que perillas u otros sensores, por ejemplo la perilla de volumen de una radio tengan por detrás un encoder rotatorio que entrega un resultado en Gray Code. Esto significa que si tienes el volumen en 10_{10} y lo subes a 11_{10} , el output del encoder cambiaría de 1111_G a 1110_G . Como ese output debe ser procesado por un microcontrolador que utiliza representación en base 2, es necesario un circuito que se encargue de hacer la traducción.

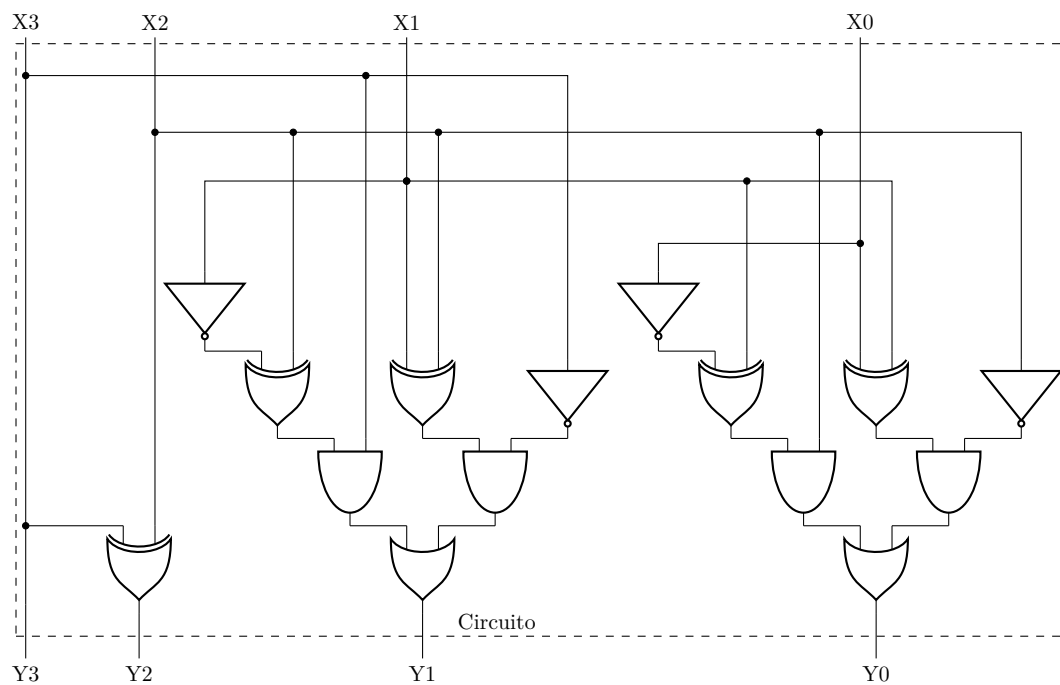
Solución:

$$Y_3 \leq X_3$$

$$Y_2 \leq X_3 \text{ xor } X_2$$

$$Y_1 \leq ((\text{not } X_3) \text{ and } (X_2 \text{ xor } X_1)) \text{ or } (X_3 \text{ and } (X_2 \text{ xor } (\text{not } X_1)))$$

$$Y_0 \leq ((\text{not } X_2) \text{ and } (X_1 \text{ xor } X_0)) \text{ or } (X_2 \text{ and } (X_1 \text{ xor } (\text{not } X_0)))$$



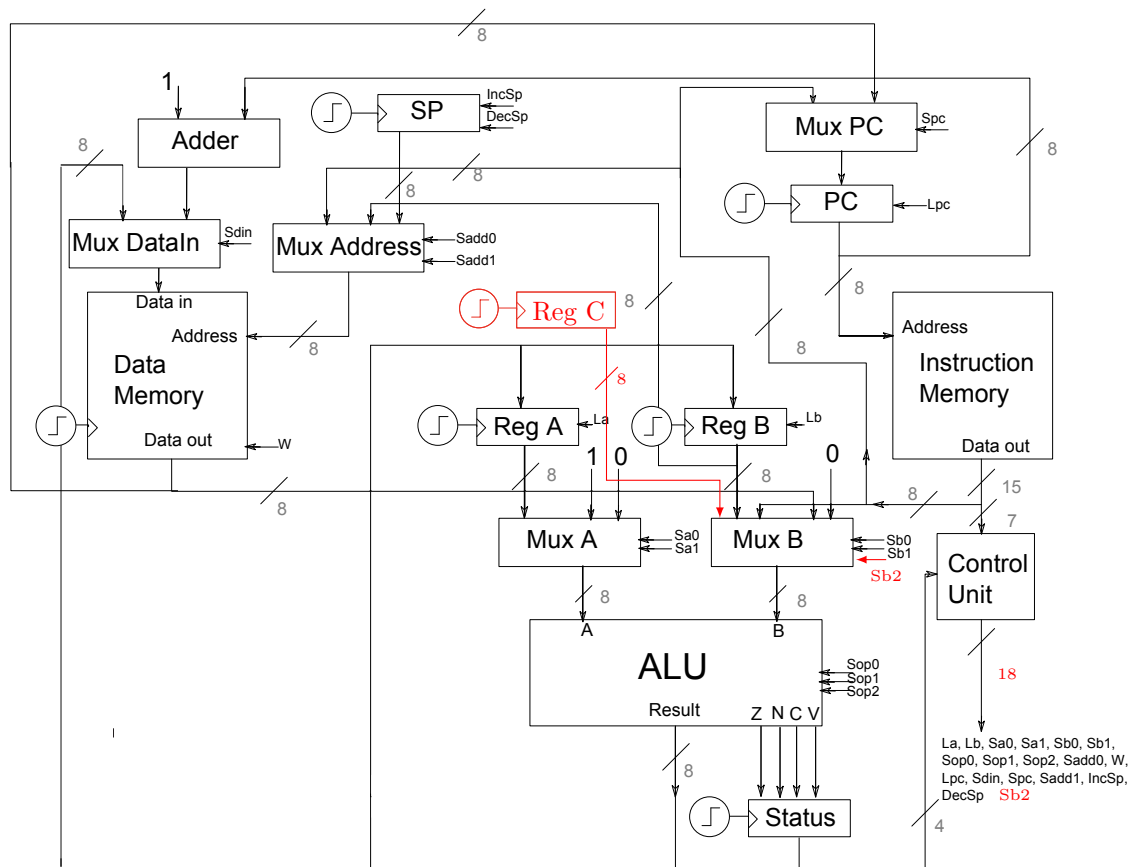
Pregunta 4: B# 2.0 (20 ptos.)

Para continuar con esta línea de investigación, se requiere agregar las siguientes instrucciones al computador básico. Para cada una indique si es posible agregarla sin modificar el *hardware* del computador. En caso de que se requiera modificar el computador, realice los cambios en el diagrama adjunto. Si corresponde, especifique las nuevas señales de control agregadas e indique el valor que deben tomar cada una de las señales de control para ejecutar cada instrucción.

Considerando un nuevo registro C de uso general:

- (a) Agregue la instrucción **ADD A,C**, la cual permite guardar en el registro A el valor de la suma del registro A y el registro C. (5)

Solución:



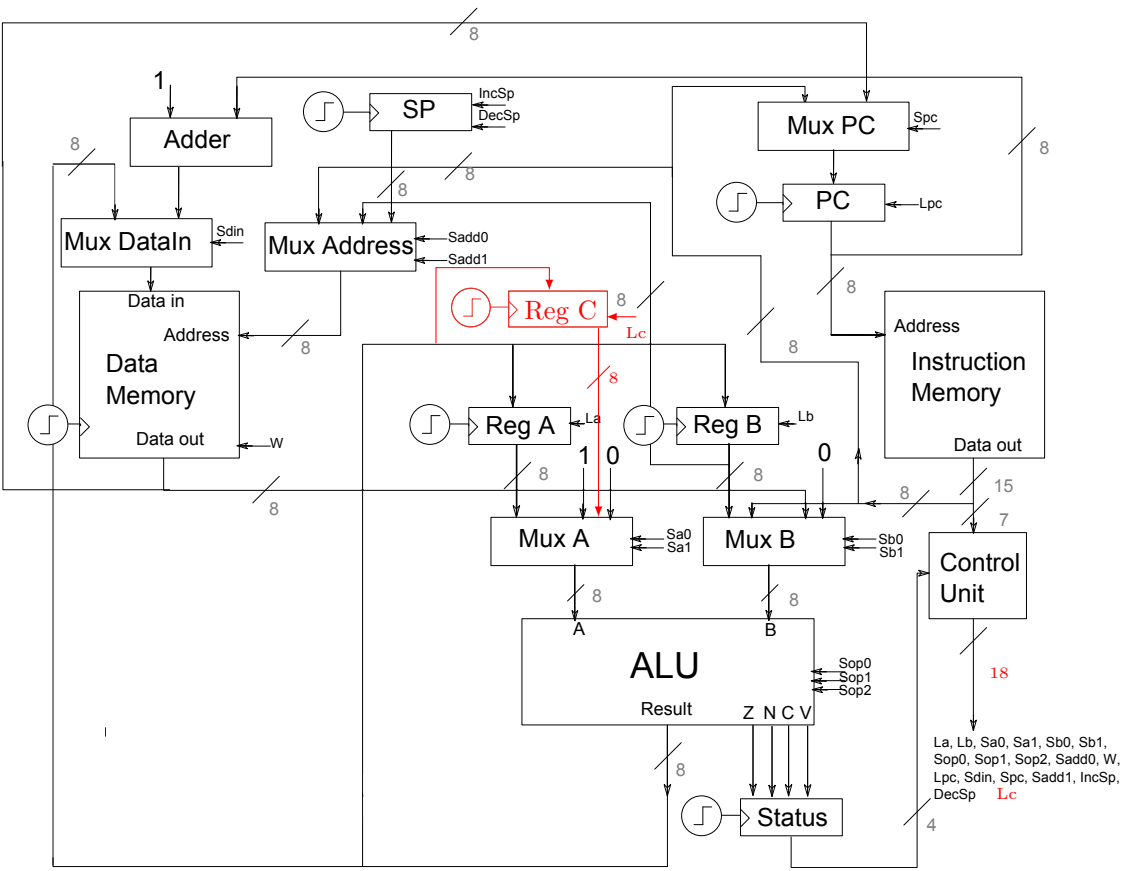
Para operar entre el nuevo registro C y el registro A, debo poder seleccionar el registro C en el Mux B, como este muxer ya no le quedan entradas debo agrandar su selector agregando la señal **Sb2**. Con eso, para la instrucción **ADD A,C** las señales de control son:

Instrucción	Operandos	Condition	Lpc	La	Lb	Sa0,1	Sb0,1,2	Sop0,1,2	Sadd0,1	Sdin	Spc	W	IncSp	DecSp	Lit
ADD	A,C	-	0	1	0	A	C	ADD	-	-	-	0	0	0	-

(b) Agregue la instrucción ADD C,Lit, la cual permite guardar en el registro C el valor de la suma del registro C y el literal.

(5)

Solución:



Para operar entre el nuevo registro C y el literal, debo poder seleccionar el registro C en el Mux A, pero en este caso no necesito agrandar el selector. Para poder guardar el resultado en el nuevo registro C debo conectar la salida de la ALU a la entrada de este y agregar la señal carga Lc. Con eso, para la instrucción ADD A,C las señales de control son:

Instrucción	Operandos	Condition	Lpc	La	Lb	Lc	Sa0,1	Sb0,1	Sop0,1,2	Sadd0,1	Sdin	Spc	W	IncSp	DecSp	Lit
ADD	C,Lit	-	0	0	0	1	C	LIT	ADD	-	-	-	0	0	0	-

- (c) Agregue la instrucción **NEG B, B**, la cual calcula el complemento de dos del registro B y lo almacena en el registro B. (5)

Solución:

Para esta instrucción no necesito hacer cambios en el diagrama, se puede realizar restándole B a 0. Las señales de control son:

Instrucción	Operandos	Condition	Lpc	La	Lb	Sa0,1	Sb0,1	Sop0,1,2	Sadd0,1	Sdin	Spc	W	IncSp	DecSp	Lit
NEG	B,B	-	0	0	1	ZERO	B	SUB	-	-	-	0	0	0	-

- (d) Agregue la instrucción **NEG A,A**, la cual calcula el complemento de dos del registro A y lo almacena en el registro A. (5)

Solución:

Para esta instrucción no necesito hacer cambios en el diagrama, se puede realizar con dos instrucciones: **NOT A,A** y **ADD A,1**. Las señales de control son:

Instrucción	Operandos	Condition	Lpc	La	Lb	Sa0,1	Sb0,1	Sop0,1,2	Sadd0,1	Sdin	Spc	W	IncSp	DecSp	Lit
NOT	A,A	-	0	1	0	A	-	NOT	-	-	-	0	0	0	-
ADD	A,Lit	-	0	1	0	A	Lit	ADD	-	-	-	0	0	0	1