Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación


**IIC2343 – Arquitectura de Computadores (II/2014)**

# Solución Interrogación 1

## Pregunta 1

a) ¿Cuál es el problema del *shift right* con los números binarios con signo?¿Es posible solucionarlo? **(1 pto.)**
**Solución:** Al interpretar el shift right como división por 2 en números binarios negativos, el signo se pierde ya que se agrega un cero a la izquierda. Esto se puede solucionar agregando explicitamente 1s en vez de 0s cuando se ejecuta esta operación.

b) ¿Que relación existe entre el tamaño de los elementos de memoria y su velocidad? Explique claramente su respuesta. **(1 pto.)**
**Solución:** Como todos los elementos electrónicos, mientras mayor es su velocidad, mayor es el costo por unidad de memoria (mega, giga, tera, etc). Por este motivo, los elementos de memoria más veloces son vendidos en tamaños mucho más pequeños que los elementos de memoria más lentos.

c) Describa el valor decimal del número 0x94A6, si este se interpreta como binario con signo. **(1 pto.)**
**Solución:** 0x94A6 = 1001010010100110 $\Rightarrow$ $C_2(1001010010100110) = 0110101101011010 = 27482$. Luego, el valor buscado es -27482

d) ¿Es posible utilizar una ALU de 8 bits, como la vista en clases, para realizar en paralelo dos sumas de números de 4 bits? Si la respuesta es positiva, describa el proceso. Si la respuesta es negativa, explique el motivo. **(1 pto.)**
**Solución:** Si asumimos que 2 números de 4 bits entran por cada bus y que la división lógica (no física) de estos se hace usando los 4 bits más significativos para uno de los números y los 4 menos siginificativos para el otro, la ALU calculará en paralelo la suma de los dos números de manera correcta, a menos que existe un carry entre el bit 3 y el bit 4, que es la posición donde se dividen los números. Este carry es en realidad overflow para los número de 4 bits, por lo que no debería tomarse en cuenta para el cálculo.

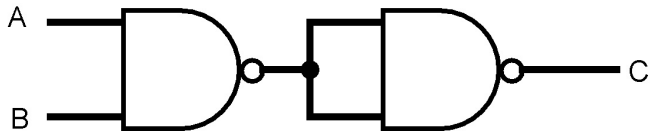
e) Describa un procedimiento para discretizar la señal bidimensional continua generada cuando se escanea un fotografía. **(1 pto.)**
**Solución:** Para realizar este proceso, dado que es una señal sin componente temporal, basta con discretizar sólo dos cantidades. La primera es la dimensión de la imagen, que se realiza dividiendo el alto y ancho de estas en pequeñas unidades indivisibles, llamadas pixels (picture elements). Luego, el valor de color de cada canal de uno de estos pixels es cuantizado, generalmente entre 0 y 255. De esta manera, una imagen digital consiste en una serie de pixels ordenados, donde cada uno almacena 3 valores enteros entre 0 y 255.

f) ¿Qué tipos de endianness se ocupan para leer datos mayores a 1 byte desde los **registros**? **(1 pto.)**
**Solución:** Los registros no utilizan endianness de ningún tipo, ya que su tamaño no está dividido en palabras de memoria. El endianness se utiliza exclusivamente en elementos de memoria que almacenan una palabra de memoria por cada dirección.

## Pregunta 2

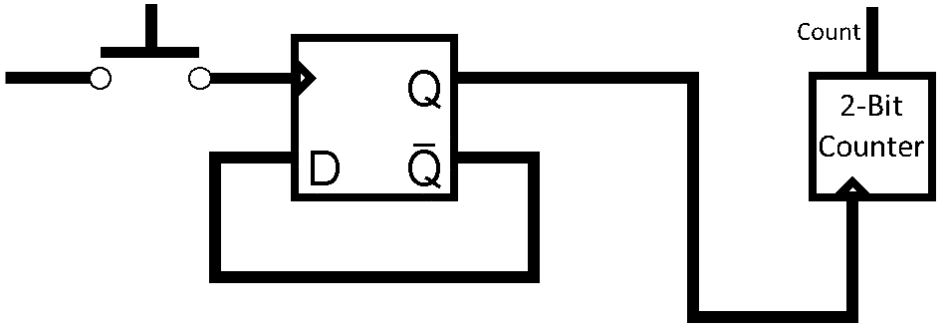
a) Implemente la compuerta AND sólo utilizando compuertas NAND (negación del AND). **(3 ptos.)**

**Solución:** Una posible solución se presenta en la siguiente figura:




b) Diseñe, utilizando todos los elementos de circuitos lógicos vistos en clases que necesite, un contador secuencial circular ascendente de 2 bits, que se incrementa cada **dos flancos** de subida de la señal de control. **(3 ptos.)**

**Solución:** La siguiente solución asume que el Flip-Flop parte con un 1 almacenado:



## Pregunta 3

a) Demuestre que los números de punto flotante del tipo *float* del estándar IEEE754 **no** cumplen el principio de asociatividad de la suma, *i.e.*, $x + (y + z) = (x + y) + z$. **(3 ptos.)**

**Solución:** Los números del tipo float tienen problemas cuando las sumas incluyen números extremadamente dispares, *i.e.*, modulos muy grandes y muy chicos. Esto se debe a que el proceso de suma implica igualar los exponentes, proceso que puede llevar a que se trunquen elementos del número de menor módulo. Teniendo esto en cuenta, si se seleccionan los valores $x = -1{,}5 \times 10^{38}, y = 1{,}5 \times 10^{38}, z = 1{,}0$, tenemos lo siguiente en el lado izquierdo de la igualdad:

$$x + (y + z) = -1{,}5 \times 10^{38} + (1{,}5 \times 10^{38} + 1{,}0) = -1{,}5 \times 10^{38} + 1{,}5 \times 10^{38} = 0$$

En este caso, al igualar los exponentes en la suma de la derecha, el valor 1,0 se pierde, ya que el significante no tiene la cantidad de cifras necesarias. Si analizamos ahora el lado derecho de la igualdad tenemos:

$$(x + y) + z = (-1{,}5 \times 10^{38} + 1{,}5 \times 10^{38}) + 1{,}0 = 0 + 1{,}0 = 1{,}0$$

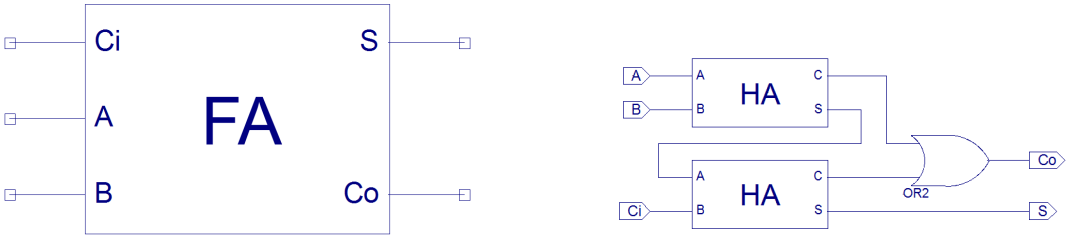
A diferencia del caso anterior, acá no se pierde información, ya que la primera operación se realiza con números con exponente similar.

b) Si hay algún problema eléctrico, como un alza de voltaje, es muy fácil corromper datos almacenados en binario. Por ejemplo, el número 10 (1010b) puede transformarse en 14 (1110b), con tan sólo modificar un bit. Describa una codificación binaria para los números 0 y 1, de manera que ésta permita detectar y corregir errores de a lo sumo 1 bit, *i.e.*, un bit de la codificación se ve alterado. **(3 ptos.)**

**Solución:** Del enunciado se obtienen claramente, que codificar un número usando sólo un bit no es suficiente para corregir errores. Si tomamos 2 bits y representamos el 0 como “00” y el 1 como “11”, también nos encontramos con un problema, ya que con un error, la codificación del 0 puede quedar como “01”, mientas que la del 1 podría quedar también como “01”. Siguiendo el mismo análisis, es fácil notar que si usamos tres bits y codificamos al 0 como “000” y al 1 como “111”, ningún error de 1 bit será suficiente para confundir las codificaciones.

## Pregunta 4

Para todas las preguntas de esta sección, considere el componente de la Figura 1a) y su versión detallada en la Figura 1b):




(a) Full Adder (b) Full Adder detallado

Figura 1

a) Construya la tabla de verdad del Full Adder, considerando las entradas A, B y Ci, y las salidas Cout y S. **(1 pto.)**

**Solución:** Una posible solución se presenta en la siguiente tabla:

| Ci | A | B | Cout | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

b) ¿Qué proceso(s) en necesario realizar en Xilinx para que un componente recién fabricado esté disponible para usarse en nuevos esquemáticos, como por ejemplo, en el Main? **(1 pto.)**

**Solución:** Para que un componente esté disponible para su uso se deben realizar los siguientes pasos en la pestaña de diseño:

- Guardar el esquemático
- Chequear las reglas del diseño (Check Design Rules)
- Crear el símbolo del esquemático (Create Schematic Symbol)

Luego de dichos pasos, el componente puede ser usado en nuevos esquemáticos.

c) Utilizando el componente del enunciado, la simbología de Xilinx y el área de trabajo adjunta, construya un sumador con tres entradas de 4 bits cada una, de forma que al procesarlo se vea como el de la Figura 2. Para esto puede basarse en la construcción realizada para el proyecto del curso. **(2 ptos.)**

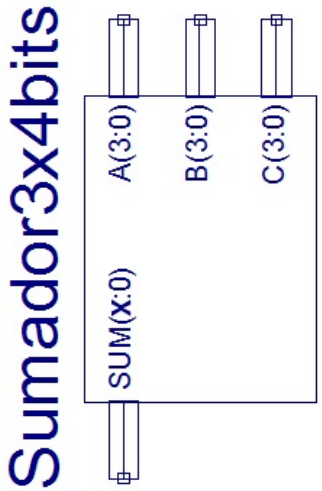
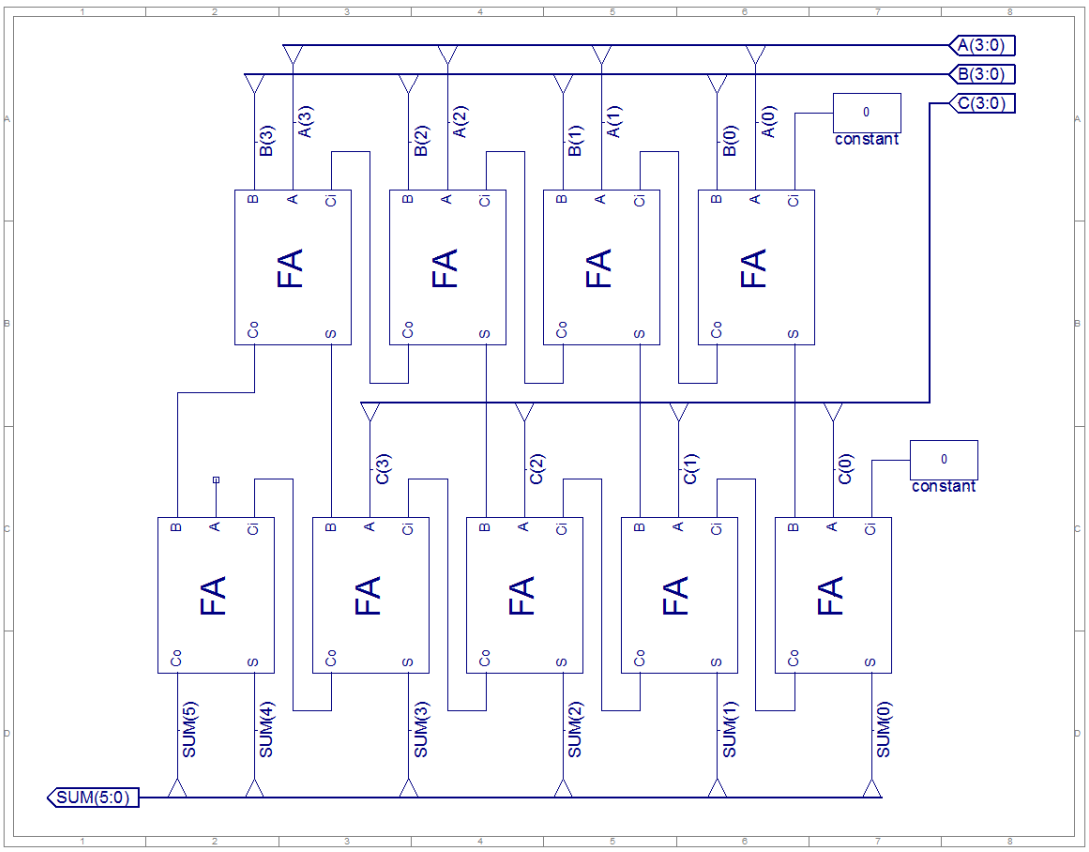



Figura 2: Sumador con tres entradas de 4 bits.

**Solución:** Una posible solución se presenta en la siguiente figura:



d) Considerando la salida SUM(X:0) del componente de la pregunta anterior, ¿cuál es el largo en bits de la palabra que sale de esta? Justifique su respuesta. **(1 pto.)**

**Solución:** El largo de la palabra es de 6 bits, pues en el peor caso $A = B = C = 1111$. Luego la suma $A + B + C = 101101$.

e) Comente, ¿qué modificaciones haría a su sumador para que ahora realizara la operación A + B – C? Incluya en su respuesta los supuestos que haga. **(1 pto.)**

**Solución:** Supongamos que $A + B > C$ y llamemos $T_0$ a la salida $S$ del Full Adder que toma $A_0$ y $B_0$, entonces para realizar la operación mencionada habría que invertir los inputs de $C$ y agregar un 1 en el $Ci$ del Full Adder que toma $C_0$ y $T_0$. De esta forma la resta se realiza correctamente.