



IIC2343 – Arquitectura de Computadores (II/2015)

Interrogación 1

Pregunta 1

- a) Describa como es posible intercambiar el valor de dos variables binarias de 4 bits, usando sólo la operación XOR. (1 pto.)

Solución: Basta con aplicar tres veces seguidas la operación XOR (\oplus), *i.e.*, $X = X \oplus Y$, $Y = X \oplus Y$, $X = X \oplus Y$.

- b) Dada una cantidad acotada de piedras (**N**), ¿qué representación posicional permite representar más números usando a lo más **N** piedras? Asuma que en cada posición, la cifra está representada por la cantidad de piedras. (1 pto.)

Solución: Siempre será la representación de base 2, ya que al usar sólo una piedra por posición, alcanza a llegar a un exponente mayor.

- c) ¿Cómo implementaría en la máquina programable una instrucción que no haga nada, *i.e.*, que no modifique el estado del computador? Describa las señales de control. (1 pto.)

Solución: Basta cualquier conjunto de señales de control, tal que L_A y L_B sean iguales a 0.

- d) Asuma que a una memoria RAM con un bus de direcciones de 12 bits, se le conecta un clock de 4KHz. ¿Cuánto tiempo toma como mínimo llenar esta memoria? (1 pto.)

Solución: Demorará 1 segundo, ya que $4K = 2^{12}$, luego en un segundo se podrán escribir 2^{12} palabras, que es justamente la capacidad de la memoria.

- e) Muestre que con la compuerta NAND se pueden representar todas las compuertas binarias de la lógica booleana. (1 pto.)

Solución: Basta con mostrar como se pueden formar AND (\wedge), OR (\vee) y NOT (\neg) usando solo NAND ($\overline{\wedge}$):

$$\neg A = A \overline{\wedge} A$$

$$A \wedge B = (A \overline{\wedge} B) \overline{\wedge} (A \overline{\wedge} B)$$

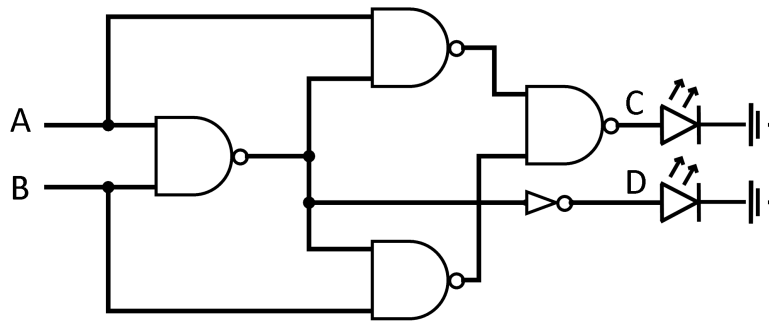
$$A \vee B = (A \overline{\wedge} A) \overline{\wedge} (B \overline{\wedge} B)$$

- f) ¿Cuántos caracteres puede tener como máximo un archivo de texto en castellano de 1KB no comprimido, donde sólo se utilizan como caracteres las letras mayúsculas, el punto, la coma y el espacio? (1 pto.)

Solución: Dado que se utilizan en total 30 caracteres, se necesitan 5 bits por carácter. Luego, la cantidad máxima de caracteres es $\lfloor \frac{1024 \times 8}{5} \rfloor = 3222$.

Pregunta 2

a) Describa con una tabla de verdad y con palabras que es lo que realiza el siguiente circuito: **(1 pto.)**



Solución: Si ambas entradas son 0, los LEDs permanecen apagados. Si sólo una de las entradas está en 1, se prende el LED superior. Finalmente, si ambas entradas están prendidas, se prende el LED inferior.

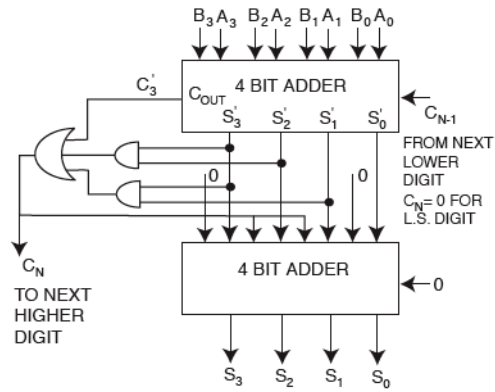
A	B	C	D
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

b) Describa como representaría en un computador, a nivel de hardware, el conjunto completo de los números racionales. Considere además de la representación, como se implementarían las operaciones aritméticas básicas (+, −, ×, /) **(2 ptos.)**

Solución: Dado que todos los números racionales se escriben como $\mathbb{Q} = \frac{a}{b}, a \in \mathbb{Z}, b \in \mathbb{N}$, es posible representarlos usando dos registros, donde uno representa el numerador y el otro el denominador. Para operarlos se utilizan las reglas básicas de los racionales:

- Multiplicación: Se multiplica numerador con numerador y denominador con denominador. Este producto se puede implementar sin problemas con compuertas lógicas.
- División: Se intercambia en el divisor, el contenido de los registros del numerador y denominador. Luego, se realiza el producto como en el caso anterior.
- Suma y resta: Se igualan los denominadores y luego se ejecuta la operación. La igualación se realiza mediante multiplicación entre números enteros y naturales, que es fácilmente implementada con compuertas lógicas, lo mismo que la posterior suma.

- c) El BCD (binary coded decimal), es una codificación que representa cada cifra de un número natural decimal utilizando una cantidad fija de bits. Utilizando las compuertas lógicas vistas en clases, diseñe un sumador que toma como entradas dos números naturales de 4 cifras, representados en BCD. **(3 ptos.)**
Solución: La siguiente figura muestra la solución para una cifra. Para 4 cifras, se usan 4 como el de la figura, conectados en cascada.



- ADD 0110 WHEN $C'_3=1$
- ADD 0000 WHEN $C'_3=0$

Pregunta 3

- a) Dada la siguiente secuencia de instrucciones de la máquina programable vista en clases: $A = 8$, $B = 5$, $A = A+B$, $A = A-B$, ¿a qué secuencia de números corresponden estas instrucciones, si son interpretadas como floats? Interprete los números usando little endian. **(1 pto.)**

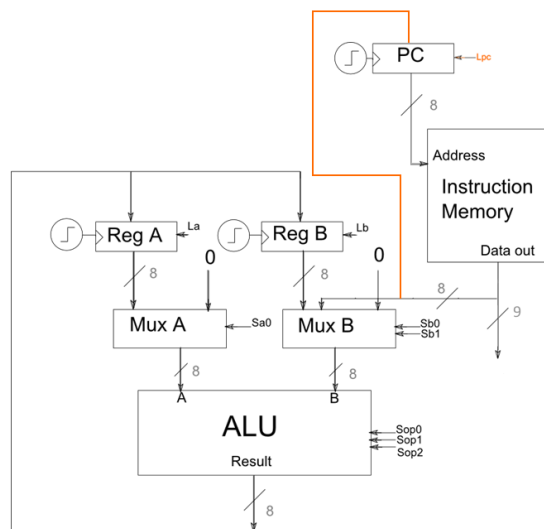
Solución: Utilizando el mismo orden que el de la máquina vista en clases, *i.e.*, L_A , L_B , S_A , S_B , ALU, más el literal a continuación, las señales de control de las instrucciones quedan de la siguiente manera: Como cada par (instrucción,literal) utiliza 16, se necesitan 2 instrucciones para poder codificar un float. El

L_A	L_B	S_A	S_B	ALU	Lit	Instr.
1	0	1	01	000	00001000	A=8
0	1	1	01	000	00000101	B=5
1	0	0	00	000	00000000	A=A+B
1	0	0	00	001	00000000	A=A-B

estándar float ordena sus bits, de más a menos significativo, de la siguiente manera: signo (1), exponente (8), significante (23). Luego, asumiendo little endian (orden de los bytes) la representación binaria y decimal de los números generados por las instrucciones son: 00000101 01101000 00001000 10101000 = $1,09101775 \times 10^{-35}$, 00000000 10000001 00000000 10000000 = $1,1846958 \times 10^{-38}$.

- b) Modifique el diagrama de la máquina programable, de manera que soporte la ejecución de la instrucción GOTO **Dir**, que fuerza que la siguiente instrucciones en ejecutarse sea la ubicada en la dirección **Dir**. **(2 ptos.)**

Solución:



- c) Asuma que dos máquinas programables de 8 bits están conectadas mediante un bus de datos unidireccional, donde los datos van desde la ALU de la primera (máquina **X**) hasta el registro A de la segunda (máquina **Y**). Considere además que el clock de la máquina **X** tiene el doble de frecuencia que el de **Y**. Para compensar por esta diferencia de velocidad, el clock de la máquina **X** se detiene después de **K** ciclos, durante un período de tiempo también igual a **K** ciclos, luego de lo cual vuelve a funcionar, repitiendo el proceso. Teniendo esto en consideración, diseñe el bus de datos, de tal manera que la máquina **Y** no pierda información. **(3 ptos.)**

Solución: El bus de datos se puede modelar fácilmente utilizando un arreglo de K registros (asumimos

K potencia de 2), donde la máquina **X** escribe secuencialmente en ellos en cada ciclo, desde 0 a $K-1$, mientras que por el lado de **Y** esta lee de estos registros también de 0 a $K-1$. Las salidas de los K registros están conectadas a un mux de K entradas por el lado de **Y**, que se encarga de seleccionar el registro a leer, y su bus de control se conecta a un contador ascendente de $\log_2(K)$ bits, que utiliza el clock de la máquina **Y**. Las entradas de los K registros están conectadas a la salida de la ALU de **X**. Para decidir en que registro escribir, *i.e.* que señal de Load activar, se utiliza un esquema similar al de las salidas, pero el contador utiliza el clock de **X** y se utiliza un demultiplexor (inverso de un multiplexor) en lugar de un multiplexor.