



EXAMEN PARCIAL DE FUNDAMENTOS DE COMPUTADORES CURSO 2014-15, PRIMER PARCIAL (CONV. DE SEPTIEMBRE), 4 DE SEPTIEMBRE 2015

1. **(1 punto)** Dados los siguientes números A: $(111000101101)_{C2}$ y B: $(100111000100)_{C2}$ codificados ambos en C2 con 12 bit, se pide:
- (0,6 puntos)** Calcular $(A+B)$ y $(A-B)$ en la misma representación indicando en cada caso si hay acarreo y/o desbordamiento.
 - (0,2 puntos)** Expresar los resultados obtenidos en decimal y octal.
 - (0,2 puntos)** ¿Cuáles de los resultados obtenidos son correctos?
2. **(1 punto)** Usando un sumador binario de 8 bits y el menor número de puertas lógicas, implemente un sistema combinacional con 2 entradas de 8 bits por las que llegan 2 números naturales, A y B, codificados en binario y con 3 salidas de 1 bit que respectivamente indican si $A > B$, si $A = B$ y si $A < B$.

3. **(2,5 puntos)** Una avioneta tiene dos luces de seguridad: {Peligro, Emergencia}.

La luz de Peligro se encenderá sólo si ocurre alguno de estos casos:

- Altura no mayor que 1500 pies, sin sistema de navegación y sin visibilidad.
- Altura mayor que 1500 pies y con fallo en el motor.
- Altura mayor que 1500 pies y sin visibilidad.

La luz de Emergencia se encenderá sólo si ocurre alguno de estos otros casos:

- Altura no mayor que 1500 pies y con fallo en el motor.
- Altura mayor que 1500 pies, sin sistema de navegación y sin visibilidad.
- Altura mayor que 1500 pies, sin sistema de navegación y con fallo en el motor.

Se pide:

- (1 punto)** Obtener la tabla de verdad del sistema, indicando la codificación elegida para las entradas y salidas.
- (1 punto)** Utilizando el menor número de puertas AND, OR y/o NOT, implementar un circuito con el comportamiento anteriormente especificado.
- (0,5 puntos)** Calcular el retardo de propagación y el retardo de contaminación del circuito atendiendo a los siguientes parámetros:

Puerta	NOT	AND (2 entradas)	AND (3 entradas)	OR (2 entradas)	OR (3 entradas)
Retardo (ps)	100	180	206	171	184

4. **(3 puntos)** Una máquina secuencial tipo Moore posee dos entradas $x, y \in \{0, 1\}$ y una salida $z \in \{0, 1, 2\}$. Suponiendo que las secuencias que recibe por sus entradas hasta el instante $t-1$ se interpretaran como números binarios sin signo:

$$N_x = (x(t-1), x(t-2), \dots, x(1), x(0))$$

$$N_y = (y(t-1), y(t-2), \dots, y(1), y(0))$$

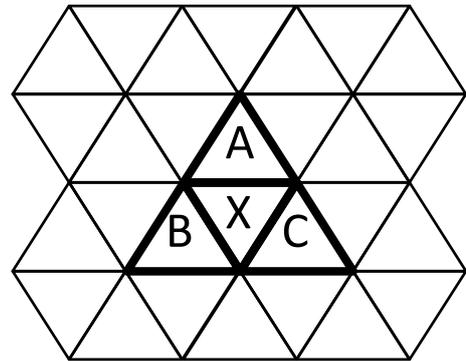
donde $x(t-1)$ e $y(t-1)$ son los bits más significativos, el sistema debe comportarse de la siguiente forma:

$$z(t) = \begin{cases} 0 & \text{Si } N_x = N_y \\ 1 & \text{Si } N_x < N_y \\ 2 & \text{Si } N_x > N_y \end{cases}$$

Se pide:

- (1,5 puntos)** Especificar el sistema mediante un diagrama de estados.
 - (0,5 puntos)** Indicar las tablas de verdad que especifican las funciones de salida y transición de estados del sistema.
 - (1 punto)** Implementar el sistema mediante biestables D y puertas NAND.
- 5. (2,5 puntos)** El juego de la vida, diseñado por el matemático británico John Horton Conway en 1970, es un juego de cero jugadores ya que su evolución está determinada por el estado inicial y no necesita ninguna entrada de datos posterior. En nuestra versión, el tablero de juego es una malla formada por triángulos que representan células (véase figura); de forma que cada célula, X, tiene 3 células vecinas A, B y C. Cada célula puede estar en uno de los siguientes estados: muerta (00), infantil (01), adulta (10) o anciana (11). El estado de cada célula depende del estado de sus vecinas y evoluciona por cada ciclo de reloj de la siguiente manera:

- Una célula muerta con 1 ó 2 células vecinas vivas nace (estará en su infancia).
- Una célula viva no anciana con 1 ó 2 células vecinas vivas sigue viva; en otro caso muere. Si sigue viva se comporta como sigue:
 - Si es infantil, pasa a adulta.
 - Si es adulta y tiene 1 vecina viva, sigue adulta.
 - Si es adulta y tiene 2 vecinas vivas, pasa a anciana.
- Toda célula anciana muere.



Se desea diseñar un circuito que reproduzca el comportamiento de una célula X en función del estado de sus células vecinas A, B y C. El circuito tendrá 3 entradas de 2 bits por las que recibirá el estado de cada una de las células vecinas y 1 salida de 2 bits por la que indicará su estado.

Se pide:

- (0,5 puntos)** Diseñar un circuito combinacional que indique el número de células vecinas a X que están vivas usando un sumador de 1 bit y el menor número de puertas lógicas.
- (1 punto)** Especificar mediante un diagrama de estados de tipo Moore el comportamiento de la célula X. El sistema tendrá como entrada la salida del anterior circuito y como salida el estado de X.
- (1 punto)** Implementar la anterior especificación usando un contador modulo-4 y el menor número de puertas lógicas.