

Grado en Ingeniería de Tecnologías de Telecomunicación. Curso 2018/2019. 2º curso.
Examen final de recuperación de Electrónica Digital I.
Santander, 31 de Enero de 2019.

1º. (1 punto) Un código de Hamming se forma añadiendo al código original una serie de bits de paridad. El código sirve para recuperar errores simples en los bits de cada palabra. Los bits de una palabra de un código se ordenan de izquierda a derecha por columnas, desde 1 en adelante (1, 2, 3, etc). En las posiciones que sean potencia de 2 se sitúan los bits de paridad P0, P1, ..., y en las que no lo sean los bits de datos D0, D1, Para una palabra de 4 bits (D3D2D1D0) el código de Hamming se forma usando la siguiente figura, donde las marcas indican que bits se usan para formar cada bit de paridad.

	1	2	3	4	5	6	7	
	P0	P1	D3	P2	D2	D1	D0	
×			×		×		×	F0 (1)
		×	×			×	×	F1 (2)
				×	×	×	×	F2 (4)

Así, la fila F0 indica que P0, D3, D2 y D0 tienen que tener paridad par, la fila F1 indica que P1, D3, D1 y D0 tienen que tener paridad par y la fila F2 que P2, D2, D1 y D0 tienen que tener paridad par. De esta forma conocidos D3, D2, D1 y D0 se generan P2, P1 y P0 (se añade 1 ó 0 de forma que el número de 1s en la fila sea 0 o par).

Al leer una palabra del código de Hamming se puede recuperar un error en uno de los bits en la palabra comprobando que los bits marcados de las filas F2 (P2, D2, D1 y D0), F1 (P1, D3, D1, D0) y F0 (P0, D3, D2, D0) tienen paridad par o no. Así se forma un número en binario $F = (F2F1F0)_2$ donde cada bit de F es 1 si su paridad es incorrecta y 0 si es correcta. El valor decimal de F es la columna que contiene el error. Si F es 0 es que no hay error. Localizado el bit erróneo se sustituye por el bit correcto, se eliminan los bits de paridad y se obtiene el dato final.

Encontrar el mensaje sin errores de datos de las siguientes cuatro palabras de un código de Hamming, con los bits ordenados como (P0P1D3P2D2D1D0) e indicar los valores de D (D3D2D1D0) sin error en binario y en hexadecimal:

- 1100010
- 0010011
- 0010111
- 1101001

2º. (1 punto) Simplificar la siguiente función lógica usando únicamente los postulados y teoremas del álgebra de conmutación.

$$\overline{\overline{A(C + \overline{D})} + \overline{B} \cdot \overline{\overline{A}} + C} \cdot \left(\overline{D + \overline{A} C + B \overline{C}} \right) + \overline{\overline{\overline{A} \overline{C} + B \overline{D}} \cdot \overline{C} + A D}$$

3º. Un circuito digital tiene cinco entradas A, B, C, D y E, y se sabe que nunca se da la situación en la que exactamente cuatro de las entradas estén a 1 ó exactamente cuatro de las entradas estén a 0. La salida Z del circuito debe fijarse a 1 lógico cuando se cumpla alguna de estas dos condiciones:

- C1. Hay exactamente dos entradas a 1, y hay al menos un 1 en las entradas A y C.
- C2. Hay exactamente tres entradas a 1, y hay exactamente dos 1s en las entradas B, C y D.

En el resto de los casos, Z será 0 lógico.

- a) Realizar una descripción VHDL de este problema directamente de su enunciado, sin realizar la tabla de verdad. (1 punto)
- b) Encontrar una forma SOP mínima para $Z = F(A, B, C, D, E)$. (1.5 puntos)
- c) Implementar la función lógica usando puertas NOR suponiendo A.L, B.H, C.L, D.H, E.H y Z.L. (0.5 puntos)

4°. a) Implementar la siguiente función lógica en cinco niveles, que desde la salida (1° nivel) hacia las entradas (5° nivel) usan puertas NOR (1° nivel), AND (2° nivel), OR (3° nivel), NOR (4° nivel) y AND (5° nivel) para entradas A.H, B.L, C.L, D.H, E.L y salida F.H. (1 punto)

$$F(A, B, C, D, E) = [B + \bar{C} + A(D + \bar{E})](A + \bar{B} + D)[C + (E + BD)(\bar{B} + \bar{E})]$$

- b) Encontrar el camino crítico del circuito, los valores necesarios en sus entradas que permiten su activación, y el tiempo de propagación máximo asociado a ese camino crítico para estos tiempos de propagación de las puertas lógicas. (0.5 puntos)

	NOT	AND2	AND3	NOR2	NOR3	OR2	OR3
tphl (ns)	10	13	15	12	14	14	18
tplh (ns)	11	14	16	13	15	13	17

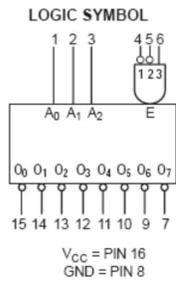
5°. a) Dado un dígito D (de 0 a 9) código BCD con pesos (5, 4, 2 -1), suponiendo sólo una codificación por dígito, realizar un circuito que realice la conversión de D al dígito correspondiente en código NBCD (pesos 8, 4, 2, 1) usando dos circuitos 74LS138 (decodificadores de 3 a 8) y un circuito 74LS147 (codificador con prioridad de 10 a 40), y puertas NOT. Las salidas y las entradas deben estar en lógica positiva. (1 punto)

b) Suponer que se añade un bits de signo S al código BCD anterior de forma que, si el bit de signo es 0 el número es positivo (+D), y si el bit el de signo es 1 el número es negativo (-D). Los números así codificados están entre +9 y -9, con dos ceros +0 y -0, aunque en este diseño se debe suponer que el -0 no es válido. Diseñar un circuito que convierta estos dígitos a números binarios en complemento-2, añadiendo al circuito del apartado a), otro circuito 74LS147 (considerando sus salidas en lógica positiva) para generar los bits menos significativos de los números negativos, y seleccionando entre las salidas de números positivos y negativos mediante un circuito 74LS157 (4 2-INP MUX). (1 punto)

c) Diseñar el mismo circuito que en el apartado b), añadiendo al circuito del apartado a), un circuito basado en el sumador 74LS83 (4-INP full-adder) y puertas lógicas que realice el complemento-2 del número positivo, solo si el bit de signo está a 1. (0.5 puntos)

6°. Diseñar un flip-flop con dos entradas X-Y tal que cuando $X > Y$ se cambia el valor del flip-flop, cuando $X < Y$ el valor del flip-flop se mantiene igual, y cuando ambas entradas son 1 el flip-flop se resetea. No se permite que las dos entradas estén a 0 a la vez.

- a). Encontrar la tabla de operación, la tabla característica, y la ecuación característica del flip-flop, y realizar una implementación “clocked-latch” a partir de un “latch” S-R NAND. (0.5 puntos)
- b) Encontrar la tabla de excitación de dicho flip-flop y construir un flip-flop J-K a partir del flip-flop X-Y. (0.5 puntos)



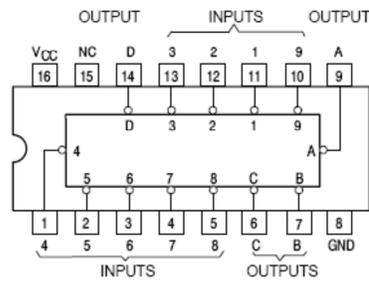
TRUTH TABLE

INPUTS						OUTPUTS							
E ₁	E ₂	E ₃	A ₀	A ₁	A ₂	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	L	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

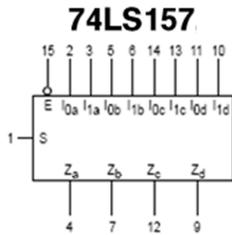
74LS138

SN74LS147 FUNCTION TABLE



INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	L	H	H	H	H	H	L	L	H
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

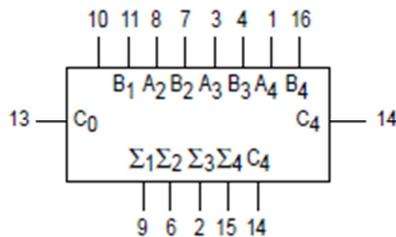
H = HIGH Logic Level, L = LOW Logic Level, X = Irrelevant



TRUTH TABLE

ENABLE	SELECT INPUT	INPUTS		OUTPUT
E	S	I ₀	I ₁	Z
H	X	X	X	L
L	H	X	L	L
L	H	X	H	H
L	L	L	X	L
L	L	H	X	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care



74LS83