Práctica 2 - Lógica Digital - Parte A

Sistemas Digitales

Segundo Cuatrimestre 2024

Todas las compuertas mencionadas en esta práctica son de 1 ó 2 entradas, a menos que se indique lo contrario. Usaremos los símbolos detallados a continuación para representar las distintas funciones lógicas: `XOR` $\rightarrow \oplus$, `NAND` $\rightarrow |$, `NOR` $\rightarrow \downarrow$

Durante la presente práctica se recomienda fuertemente la utilización de un simulador para experimentar con los componentes y circuitos propuestos y verificar las soluciones. Una recomendación es el Logisim (http://www.cburch.com/logisim/).

Circuitos Combinatorios

Ejercicio 1 Demostrar si las siguientes equivalencias de fórmulas booleanas son verdaderas o falsas:

a) $x.z = (x + y).(x + \overline{y}).(\overline{x} + z)$

b) $x \oplus (y.z) = (x \oplus y).(x \oplus z)$ donde se aplica la propiedad distributiva con respecto a $\oplus$[1]

Ejercicio 2 Una *fórmula* del álgebra de Boole es:

- $p, q, r, \ldots$, una variable booleana que puede tener valor 1 o 0,
- 1, la constante *verdadero*,
- 0, la constante *falso*,
- Si p y q son fórmulas, entonces $p + q$ (p OR q), $p.q$ (p AND q) y $\overline{p}$ (la negación de p) son fórmulas.

¿Se pueden expresar todas las funciones totales[2] $f : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$ usando fórmulas del álgebra de Boole? Justificar.

Ejercicio 3 Determinar la veracidad o falsedad de las siguientes afirmaciones:

a) Sea $p|q = \overline{p.q}$ ¿Alcanza este único operador (`NAND`) para representar todas las funciones booleanas?

b) Sea $p \downarrow q = \overline{p + q}$ ¿Alcanza este único operador (`NOR`) para representar todas las funciones booleanas?

Ejercicio 4 Dibujar circuitos que implementen las siguientes funciones booleanas:

a) $f(A, B, C) = A.B.C$ usando 2 compuertas NOR y varias compuertas NOT.

b) $f(A, B) = \overline{(A.B) + (\overline{B}.A)}.\overline{B}$ ¿Para qué valores de A y B la función devuelve un 1?

c) $f(A, B, C, D) = ((C \downarrow D) \oplus (B + A)).(((A|B)|C)) + (\overline{D}.B))$

Ejercicio 5 Dadas las funciones booleanas F y G definidas a partir de las siguientes tablas de verdad:

[1] $p \oplus q = (\overline{p}.q) + (p.\overline{q})$.

[2] Una función total es aquella para la que todo elemento del dominio tiene imagen.

A	B	C	$F(A,B,C)$
1	1	0	0
1	0	0	1
1	0	1	1
0	1	0	0
0	0	1	0
0	1	1	1
1	1	1	1
0	0	0	0

D	E	F	$G(D,E,F)$
0	0	0	1
0	1	0	0
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	1
1	0	1	0
1	1	1	1

a) Escribir la *suma de productos* para ambas funciones. Calcular la cantidad de compuertas que la implementación literal requeriría en cada caso.

b) ¿Se pueden simplificar las expresiones usando propiedades del álgebra booleana? Para cada función decidir si es posible y, en caso de que lo sea, dibujar el circuito utilizando la menor cantidad de compuertas que pueda.

Ejercicio 6 Armar un circuito que invierta o no tres entradas de acuerdo al valor de una entrada adicional que actúa como control. En otras palabras, un inversor de k-bits es un circuito de $k+1$ entradas $(e_k, ..., e_0)$ y k salidas $(s_{k-1}, ..., s_0)$ que funciona del siguiente modo:

- Si $e_k = 1$, entonces $s_i = \overline{e_i} \qquad \forall i < k$
- Si $e_k = 0$, entonces $s_i = e_i \qquad \forall i < k$

Ejemplo:

inversor(1,011)=100 inversor(0,011)=011
inversor(1,100)=011 inversor(1,101)=010

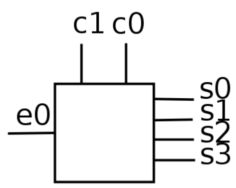
Ejercicio 7

a) Diseñar un componente con 4 entradas $e_0, \ldots, e_3$ y 4 salidas $s_0, \ldots, s_3$ que calcule el inverso aditivo del número codificado en complemento a 2 por la entrada.

b) Modificar el circuito anterior para que en una nueva salida indique si el número de la entrada no tiene un inverso aditivo representable con 4 *bits* en complemento a 2.

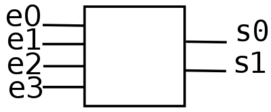
Ejercicio 8 Dibujar el diagrama lógico de un *demultiplexor* de 2 líneas de control, 1 línea de entrada y 4 líneas de salida. Este circuito dirige la única línea de entrada a una de cuatro líneas de salida, dependiendo del estado de las dos líneas de control.

c_1	c_0	s_i
0	0	$s_0 = e_0$, $s_i = 0$ si $i \neq 0$
0	1	$s_1 = e_0$, $s_i = 0$ si $i \neq 1$
1	0	$s_2 = e_0$, $s_i = 0$ si $i \neq 2$
1	1	$s_3 = e_0$, $s_i = 0$ si $i \neq 3$



Ejercicio 9

a) Dibujar el diagrama lógico de un *codificador* de 4 líneas de entrada (e_i)y 2 líneas de salida (s_i). Si únicamente e_i está alta, las salidas deben representan el número i en notación sin signo. No está definido cuál es el resultado si no se cumple que sólo una de las líneas de entrada tiene valor 1.



b) Dotar al circuito anterior de una salida adicional que indique si el estado de la entrada es válido o inválido.

Ejercicio 10

a) Dibujar con compuertas lógicas el circuito de un *decodificador* de 2 líneas de entrada (e_i)y 4 líneas de salida (s_i), cuya tabla de verdad es la siguiente:

e_1	e_0	s_3	s_2	s_1	s_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

b) Usando el circuito anterior, reescribir el *demultiplexor* de 1 línea de entrada, 2 líneas de control y 4 líneas de salida.