Introducción al Diseño Lógico 2024
Guía de Trabajos Prácticos Nº2

Ejercicio Nº1

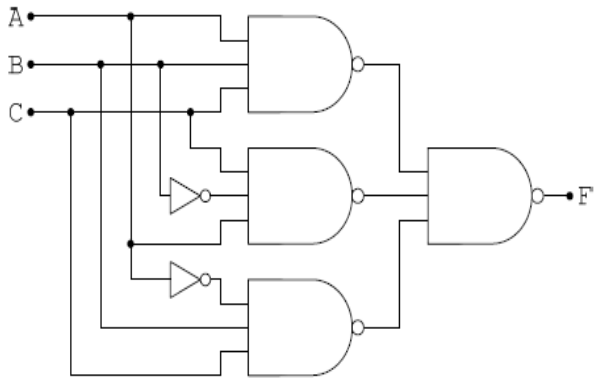

Figura 1

a) Simplifique el circuito de la Fig. 1 utilizando el álgebra de Boole.
b) Obtenga la tabla de verdad correspondiente y confeccione el diagrama de Karnaugh. A partir del mismo, obtenga una expresión lógica y compare con la obtenida en el punto anterior.

Ejercicio Nº2

TABLA 1

A	B	C	F
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**0**

a) Diseñe el circuito lógico que está representado por la Tabla 1.
b) Modifique la implementación del circuito para que esté formado sólo por compuertas de tipo NAND. De ser posible, simplifíquelo, pero sin emplear otro tipo de compuerta.
c) Modifique la implementación del circuito para que esté formado sólo por compuertas de tipo NOR. De ser posible, simplifíquelo, pero sin emplear otro tipo de compuerta.

Ejercicio Nº3

Se desea activar una señal de alarma /ALRM cuando se intente encender el motor de un auto y alguno de los dos ocupantes del asiento delantero esté sentado y no haya abrochado su cinturón de seguridad.

Para ello se cuenta con dos sensores de presencia que activan las señales /CONDUCTOR y /ACOMPAÑANTE activas en bajo cuando hay una persona ubicada en los asientos respectivos. Además se deben tener en cuenta las señales CINTUR_COND y CINTUR_ACOM que se activan en alto cuando los cinturones de seguridad respectivos han sido abrochados. También hay una señal CONTACTO activa en alto que indica si se activó la llave de encendido.

a) Realice el diagrama de Karnaugh para la generación de la señal /ALRM. A partir de lo obtenido, diseñe el circuito lógico necesario para generar la señal de alarma.

b) Implemente solo con compuertas NAND.

c) Ahora implemente solo con compuertas NOR.

Ejercicio Nº4

Diseñe un circuito cuya salida esté en nivel ALTO sólo cuando las tres entradas posean el mismo nivel.

a) Utilice un mapa de Karnaugh para obtener una expresión y un circuito simplificados a partir de la representación de tipo SOP.

b) A partir de la observación de la expresión y/o del mapa de Karnaugh que obtuvo, ¿podría modificar la implementación del circuito para que utilice una menor cantidad de compuertas?

Ejercicio Nº5

La Fig. 2 muestra un multiplicador binario de dos bits (X = x1 x0) , (Y = y1 y0). Diseñe su circuito lógico teniendo en cuenta que el resultado se encuentra expresado en cuatro bits (Z = z3 z2 z1 z0).

Figura 2

Ejercicio Nº6

Repita el diseño del problema anterior considerando que X e Y son enteros de dos bits representados en complemento a 2.

a) Realice un diseño considerando las representaciones en complemento a dos de las entradas y las salidas posibles.

b) Intente aplicar el multiplicador obtenido en el ejercicio anterior para realizar este diseño.

Ejercicio Nº7

Utilice el mapa de Karnaugh para simplificar las expresiones canónicas que representan a las siguientes tablas de verdad (Tabla 2, Tabla 3, Tabla 4, Tabla 5), según su representación como suma de productos u otra representación que incluya XOR.

Tabla 2

A	B	C	F
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**0**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**0**

Tabla 3

A	B	C	D	F
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**1**
0	0	1	1	**0**
0	1	0	0	**1**
0	1	0	1	**0**
0	1	1	0	**0**
0	1	1	1	**1**

Tabla 4

A	B	C	D	F
0	0	0	0	**1**
0	0	0	1	**1**
0	0	1	0	**X**
0	0	1	1	**1**
0	1	0	0	**X**
0	1	0	1	**X**
0	1	1	0	**1**
0	1	1	1	**1**

Tabla 5

A	B	C	D	F
0	0	0	0	**1**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	**0**
0	1	0	0	**1**
0	1	0	1	**0**
0	1	1	0	**0**
0	1	1	1	**0**

Tabla 3

A	B	C	D	F
1	0	0	0	**1**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	**1**
1	1	0	0	**0**
1	1	0	1	**1**
1	1	1	0	**1**
1	1	1	1	**0**

Tabla 4

A	B	C	D	F
1	0	0	0	**0**
1	0	0	1	**1**
1	0	1	0	**0**
1	0	1	1	**1**
1	1	0	0	**0**
1	1	0	1	**1**
1	1	1	0	**0**
1	1	1	1	**1**

Tabla 5

A	B	C	D	F
1	0	0	0	**1**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	**0**
1	1	0	0	**0**
1	1	0	1	**0**
1	1	1	0	**0**
1	1	1	1	**0**

Ejercicio Nº8

Suponga un número A representado en Ca2 de 4 bits (a3 a2 a1 a0) con LSB=a0. Se quiere diseñar un circuito lógico que provea tres salidas, una que vaya a ALTO si el número es positivo, otra que tome el valor ALTO si el número es > 3 y otra que tome el valor ALTO si el número es menor que -4.

Ejercicio Nº9

Los circuitos votadores son un elemento fundamental de muchos sistemas tolerantes a fallas basado en Triple Redundancia Modular (TMR); brevemente dicho los sistemas TMR triplican todos los recursos del sistema, y votan el resultado correcto por mayoría simple (2 de 3 al menos). De esa forma la salida del votador será siempre correcta incluso aunque uno de los sistemas que votan el resultado falle.En la Figura 3 se puede ver un ejemplo de uso de un votador digital utilizado obtener una salida V que será el resultado de lo que se presente en la mayoría de las tres entradas S_0, S_1, y S_2.

a) Escriba la tabla de verdad y escriba la función lógica que la implementa a partir de la misma.
b) Luego escriba la función logica que obtiene usando un K-map

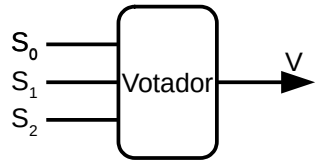

Figura 3

Ejercicio Nº10

Diseñe un detector de magnitud relativa entre números X e Y de tres bits (X = x2 x1 x0 , Y = y2 y1 y0) como el que se muestra en la Fig.4. El mismo debe poseer tres salidas que se definen de la siguiente manera:

M = 1 sólo cuando X e Y son iguales.

N = 1 sólo cuando X es mayor que Y.

P = 1 sólo cuando X es menor que Y.

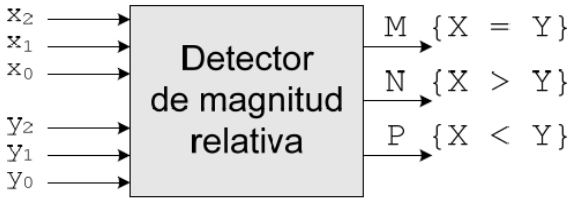

Figura 4

Ejercicio Nº11

Diseñe un multiplexor de dos entradas como el que se muestra en la Fig. 5.
Este tipo de circuito lógico funciona de la siguiente manera:

La salida S será igual a la entrada A cuando la señal de control sea C = 0.

La salida S será igual a la entrada B cuando la señal de control sea C = 1.

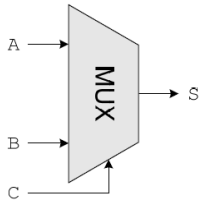

Figura 5

Ejercicio Nº12

Ahora diseñe un multiplexor de cuatro entradas como el que se muestra en la Fig. 6.
Este tipo de circuito lógico funciona haciendo que la salida S sea igual a A, B, C o D Según las entradas de control E y F como se muestra en la tabla.

a) Implemente utilizando los métodos vistos hasta ahora (tabla de verdad y diagrama de Karnaugh)
b) Implemente utilizando el multiplexor de dos entradas del ejercicio anterior.

E	**F**	**S**
0	0	**A**
0	1	**B**
1	0	**C**
1	1	**D**

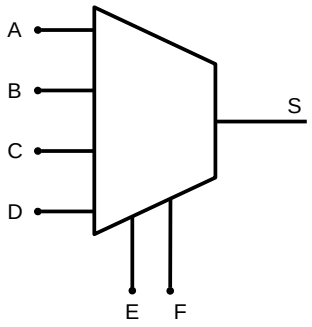

Figura 6

Ejercicio Nº13

De un conversor digital/analógico de muy alta frecuencia provienen muestras de una señal de radiofrecuencia de 5 bits con codificación signada de módulo-y-signo. El procesamiento posterior de estas muestras requiere su conversión a codificación Complemento-2.
Diseñe el circuito que toma a su entrada un número $X=X_4X_3X_2X_1X_0$ de 5 bits en codificación módulo-y-signo, y pone en su salida Y el mismo valor codificado en Complemento-2.

a) ¿Cuál es el rango de valores representable en la entrada? ¿Cuántos bits son necesarios para representar este mismo rango en la salida Y? ¿Cuántas funciones lógicas individuales es necesario generar?

b) Encuentre dichas funciones lógicas mediante diagramas de Karnaugh.

Ejercicio Nº14

Se necesita un nuevo comparador de magnitud relativa con las mismas salidas del ejercicio 10, pero las entradas son ahora números de 8 bits, $X = X_7X_6X_5X_4X_3X_2X_1X_0$, $Y = Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$.

Responda: ¿Cuántas entradas tendrán cada una de las tres funciones lógicas que es necesario sintetizar? ¿Es práctico llevar a cabo el diseño mediante Mapas de Karnaugh?

En ocasiones el proceso de diseño puede ayudarse de un poco de análisis.

Cuestión: Examine la función M, que indica igualdad. Un número será igual a otro solamente si ambos son iguales bit a bit. ¿Cómo se puede expresar esto en forma de función lógica en función de los bits de X e Y? Escriba la función lógica M que indique la igualdad.

Desgraciadamente, nuestro problema no termina en detectar la igualdad, y las comparaciones N y P son menos triviales, pero piense un segundo en la forma en la que usted compararía dos números para determinar cuál es mayor:

1. Comienzo mirando el dígito más alto de ambos.
2. Comparo los dígitos que miro.
   a) Si uno de los dos es mayor que el otro, el número correspondiente es el mayor de los dos sin importar cuánto valgan los dígitos restantes.
   b) Si en cambio ambos son iguales entonces la comparación no decide. En ese caso paso mi atención al dígito siguiente y repito desde el paso 2.
3. Repito las veces que sea necesario o hasta acabar los dígitos.En este último caso los números son iguales.

Un algoritmo semejante permite decidir la minoridad.

Aplicado al problema original, se propone que implemente un módulo de comparación de un dígito binario que tome como entradas:

- X(n) y Y(n): Dígitos binarios enésimos de los números a comparar.
- M(n-1): Vale 1 sólo si X(n-1)X(n-2)…X(0) = Y(n-1)Y(n-2)…Y(0)
- N(n-1): Vale 1 sólo si X(n-1)X(n-2)…X(0) > Y(n-1)Y(n-2)…Y(0)

Y genere como salida:

- M(n): Vale 1 sólo si X(n)X(n-1)…X(0) = Y(n)Y(n-1)…Y(0)
- N(n): Vale 1 sólo si X(n)X(n-1)…X(0) > Y(n)Y(n-1)…Y(0)

Cuestiones:

a) Diseñe el módulo de comparación de un dígito, sintetizando las funciones lógicas M(n) y N(n) a partir de las entradas X(n), Y(n), M(n-1) y N(n-1).

b) Haga un diagrama donde muestre como conectar varios de estos módulos para construir las señales M y N del comparador de números de 8 bits que necesitamos.

c) No dijimos nada de generar la señal P, pero es trivial generarla a partir de N y M. ¿Cómo lo haría?

¡Esta idea puede escalarse para comparar números de una cantidad arbitraria de bits! Pero cuidado, porque en la práctica cualquier compuerta lógica REAL tiene un tiempo de retardo tp entre que sus entradas toman un valor y que la salida toma el valor acorde a la entradas.

Cuestión: ¿Qué ocurre con el tiempo máximo que tarda este circuito en decidir correctamente la relación entre dos números si hago crecer la cantidad de bits a comparar?

Ejercicio Nº15

Los microprocesadores utilizados en aplicaciones de tipo crítica (aeronáutico, espacial, y más recientemente automotriz) frecuentemente utilizan bits de paridad asociados a cada registro interno del procesador para detectar inconsistencias en los datos ocurridas por desperfectos del sistema. Cuando se detecta esta condición, el procesador generalmente emite una excepción de error grave y pone todos los sistemas en un modo seguro (apagando el motor del auto, por ejemplo).

La verificación de esta paridad debe realizarse mediante lógica digital, ya que debe operar en paralelo con la ejecución del software del sistema y a la frecuencia de operación del sistema.

Diseñe un módulo de verificación que detecte errores de paridad en un registro de 16 bits. Las entradas al verificador son los 16 bits de un registro $R = R_{15}R_{14}R_{13}…R_{00}$, y el valor del bit de paridad par P que se calculó cuando se almacenó el valor que contiene el registro.

El módulo tiene una salida /ERROR que se pone en BAJO si se detecta una inconsistencia en la comprobación de la paridad. Si la cantidad de bits en 1 en el registro R es par, P debe valer 0.

Si la cantidad de bits en 1 en el registro R es impar, P debe valer 1.

(note que las condiciones anteriores garantizan que la cantidad de bits en uno sumados entre R y P sea siempre par, de ahí lo de “paridad par”) En cualquier otro caso significa que la paridad P no se corresponde con el valor almacenado en R, y por lo tanto el dato en R es inválido. Cuestión: Dibuje un esquema lógico que le permita generar

la señal de salida /ERROR a partir de $R_{15}R_{14}R_{13}…R_{00}$ y P

Recomendación: Es fácil ver que una aproximación vía tabla de verdad será tan tediosa como estéril. Piense cómo puede usar compuertas XOR para cumplir con el objetivo buscado.

Ejercicio Nº16

Diseñe un multiplexor de 8 entradas con habilitación:

Si la habilitación $\overline{E} = 1$, la salida **Y** permanece en nivel bajo permanente.

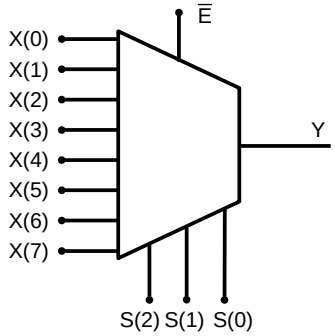

Figura 7

Si la habilitación $\overline{E}$ = 0, entonces la salida **Y** = **X(n)**, donde X(7), X(6),…,X(1),X(0) son las 8 entradas seleccionables, y n es el número formado por los tres bits S(2)S(1)S(0) del multiplexor.

Recomendación: Recicle los diseños de multiplexores de cuatro y dos entradas que ya realizó y agregue la lógica adicional que considere necesaria para resolver este problema.