

Laboratorio #03

Este laboratorio será trabajado de forma individual y se entregará de forma digital de acuerdo a la fecha de entrega en Canvas. Haga los ejercicios **SIN** usar una calculadora (a menos que se le indique lo contrario). Deberá identificar su entrega con su nombre, carné y sección. Deberá hacer un repositorio con carpetas por cada ejercicio

Ejercicio 01

Una tabla de verdad con N entradas tiene 2^N filas: una por cada posible combinación de las entradas. Cada una de estas filas está asociada a un mintermino que es **VERDADERO** para esa fila (y **sólo** para esa fila). Es posible escribir una ecuación booleana de **cualquier** tabla de verdad haciendo una **suma (or)** de todos los minterminos para los cuales la salida es **VERDADERA**. Vea, por ejemplo, la siguiente tabla de verdad:

A	B	Y	minterm	minterm name
0	0	0	$\bar{A} \bar{B}$	m_0
0	1	1	$\bar{A} B$	m_1
1	0	0	$A \bar{B}$	m_2
1	1	1	$A B$	m_3

Figure 1: Ejemplo SOP

La tabla de verdad tiene **4** minterminos en total (m_0 hasta m_3). Sin embargo los minterminos para los cuales la salida es **verdadera** son únicamente el m_1 y el m_3 . Por lo tanto una ecuación booleana que describe el funcionamiento de la tabla sería:

$$Y = m_1 + m_3 = \bar{A} \cdot B + A \cdot B$$

Esta ecuación es llamada la *suma-de-productos* o **SOP**.

Así como existe una ecuación formada por minterminos llamada SOP existe también una ecuación formada por **maxtérminos** llamada **producto-de-sumas** (o **POS**). Utilicemos la misma tabla de verdad anterior:

A	B	Y	maxterm	maxterm name
0	0	0	$A + B$	M_0
0	1	1	$A + \bar{B}$	M_1
1	0	0	$\bar{A} + B$	M_2
1	1	1	$\bar{A} + \bar{B}$	M_3

Figure 2: Ejemplo POS

Ya que este circuito es de **2** entradas tendremos un total de $2^2 = 4$ maxtérminos. La ecuación booleana **producto-de-sumas** se forma haciendo un **AND** de todos los **maxtérminos** para los cuales la salida es **FALSA**. De esta cuenta la ecuación booleana sería la siguiente:

$$Y = M_0 \cdot M_2 = (A + B) \cdot (\bar{A} + B)$$

Tanta la ecuación **SOP** como la **POS** son ecuaciones que describen la tabla de verdad, pero no necesariamente son las ecuaciones minimizadas.

Para las siguientes tablas de verdad encuentre la ecuación SOP y POS:

Tabla 01

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Tabla 02

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Tabla 03

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

Tabla 04

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

En total deberá entregar 8 ecuaciones booleanas: una SOP y una POS por cada tabla.

Ejercicio 02

Implemente una ecuación booleana de cada tabla de verdad del Ejercicio 01 en el *assignment* en CircuitVerse. En total deberá implementar 4 circuitos: 1 por tabla (usted escoge si implementa la POS o la SOP).

Tome una captura de pantalla a cada ejercicio que implemente y agregue esta captura a su entrega de laboratorio en Canvas.

Ejercicio 03

Implemente **TODAS** las ecuaciones booleanas del ejercicio 01 en verilog utilizando *gate level modelling*.

Deberá hacer cada ejercicio en un folder individual (en total necesitará 8 carpetas). Dentro de cada carpeta deberá ejecutar el comando `apio init -b TinyFPGA-BX`. Este comando creará un archivo `.ini` que incluye la configuración del **TinyFPGA**.

Después de eso deberá crear **2** archivos con extensión verilog (`.v`). El nombre del primer archivo es irrelevante, pero el segundo sí tiene que tener un formato. A manera de ejemplo digamos que usted creó un archivo llamado `tabla01POS.v`. El segundo archivo deberá tener el siguiente nombre: `tabla01POS_tb.v`. Note que lo importante es agregar el `_tb` al final. Estas letras le indican al simulador que ese archivo será el *testbench*. Para este laboratorio uno de los dos archivos de verilog deberá estar completamente vacío y su ejercicio estará en el otro archivo. No importa cuál de los dos archivos escoja. Para poder ejecutar su código deberá utilizar el comando `apio sim`.

- Puede ver un ejemplo del código (que vimos en clase) en el siguiente repositorio: https://github.com/kekellner/digital1_lab03.
- Su simulación deberá incluir pruebas para **cada fila** de la tabla de verdad (es decir, si su tabla de verdad es de 3 entradas deberá tener 8 pruebas diferentes).

En la entrega de Canvas deberá adjuntar un link a su repositorio (su repositorio debe ser público) y también deberá entregar un archivo `.zip` con todos sus códigos.

Ejercicio 04

Utilice *bubble pushing* (el teorema de DeMorgan) para re-dibujar los siguientes 2 circuitos para luego escribir la ecuación booleana del circuito por inspección sencilla.

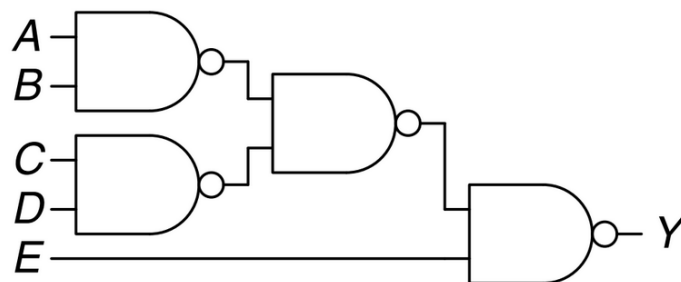


Figure 3: Ejercicio 00 de *Bubble Pushing*

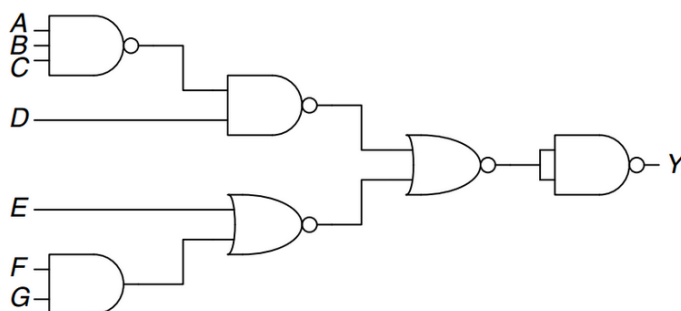


Figure 4: Ejercicio 01 de *Bubble Pushing*