Universidad Nacional Autónoma de México

Facultad de Ingeniería

División Ingeniería Eléctrica

Proyecto 1:

Función Booleana con Compuertas Lógicas NAND y NOR

Diseño Digital Moderno

Profesor:

Ing. Roberto F. Mandujano Wild

Fecha de entrega:

15/06/2021

Integrantes:

Barrera Peña Victor Miguel

Montiel Martínez Luis Javier

**Contenido**

[Introducción 2](#_Toc74679399)

[Desarrollo 2](#_Toc74679400)

[Conclusión 3](#_Toc74679401)

# Introducción

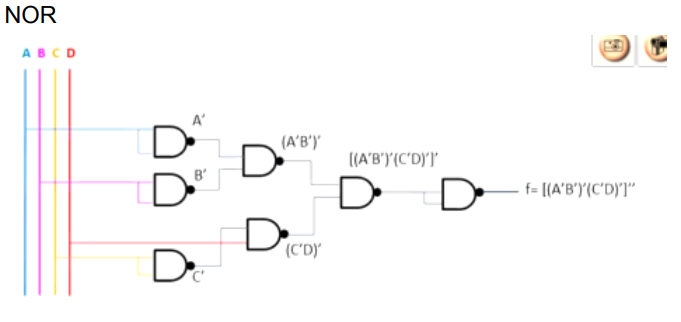
Los circuitos están presentes en nuestra vida diaria, desde el simple hecho de escribir este documento pasa por el teclado hasta almacenarse en una unidad de almacenamiento. Comprender como funciona esto es de vital importancia para el ingeniero, en computación que puede llegar a trabajar con esto directamente (hardware) o indirectamente (software); Por ello es importante saber como es que funciona, sabiendo un poco más de la implementación, es como el ejemplo de lo que sucede cuando realizamos programas para ensamblador y al final conocemos en lo que se convierten los lenguajes de alto nivel e intentamos aplicar ciertas técnicas que se implementan de manera ultra eficiente.

Después de usar compuertas es posible extraer dicho conocimiento e incluirlo que en lugar de hacer comparaciones tan exhaustivas, podemos intercambiarlo por **and’s** lógicos y de esa manera se hace un comprobación más simple.

La realización de compuerta lógicas es el primer paso hacia la elaboración de la parte física que da vida a las computadoras, es como aquella vez que se realizó el un programa muy sencillo y a finales de la carrera logras hacer herramientas tan potentes que pueden llegar a gestionar empresas, en el caso de hardware podría ser el primer paso para desarrollar tal vez sistemas de audio o incluso un procesador.

# Desarrollo

Función Original



## Tabla de verdad

Iniciaremos con la tabla de verdad en donde únicamente se realiza exponiendo las variables y el resultado de su evaluación

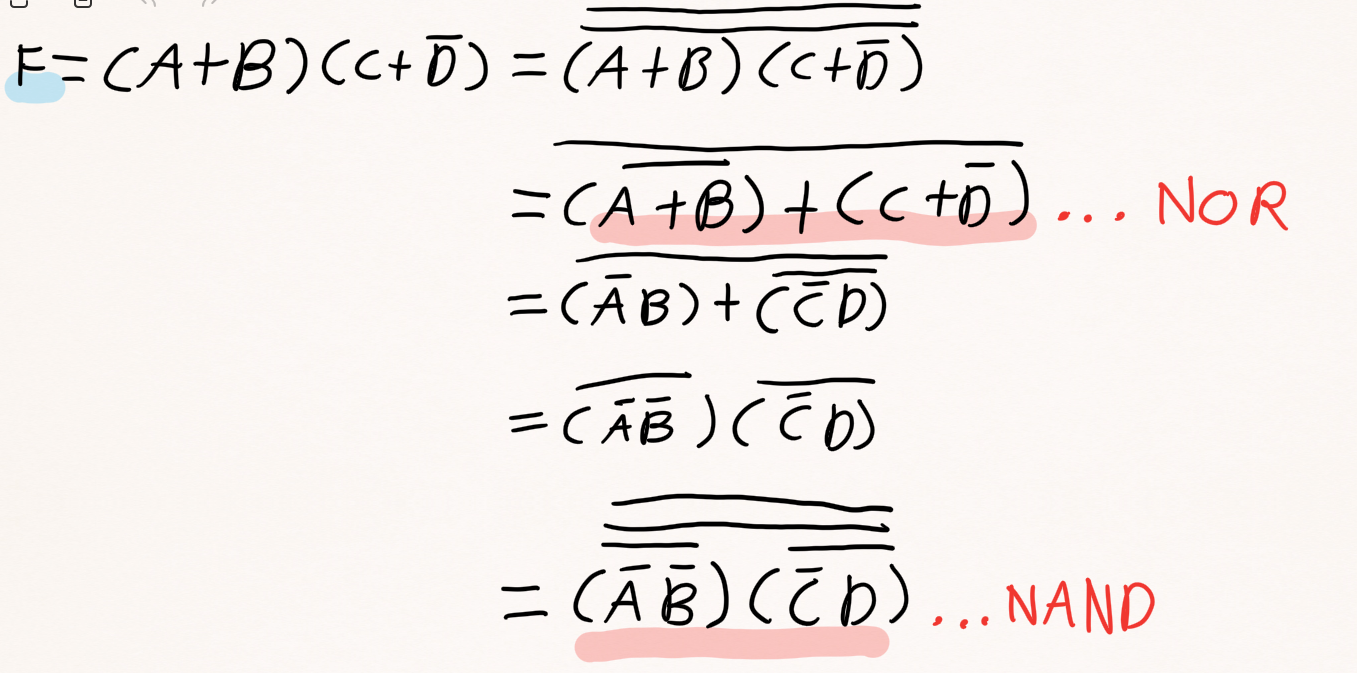
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D |  |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Función Original

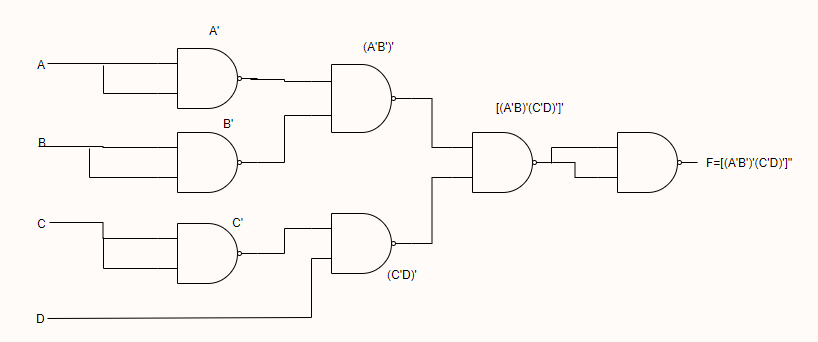
Las actividades que se tienen que realizar para ambos son:

## Desarrollo para compuertas NAND.

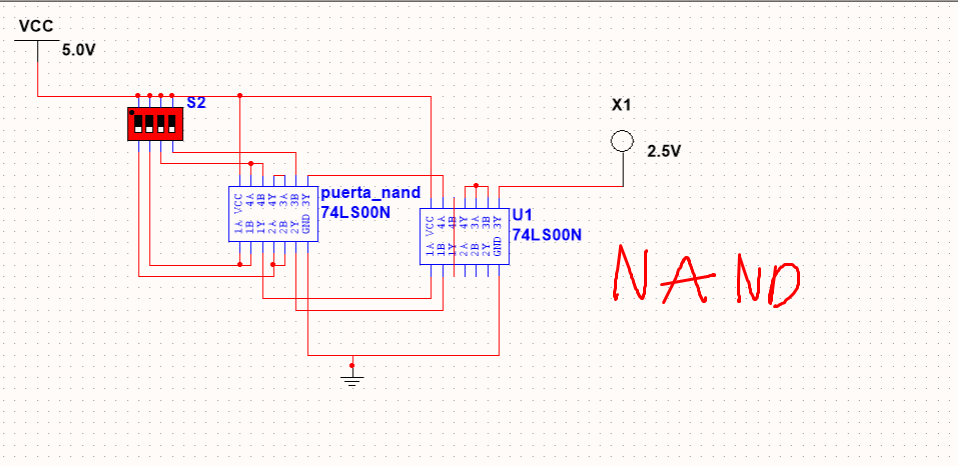
## Desarrollo compuertas NOR.



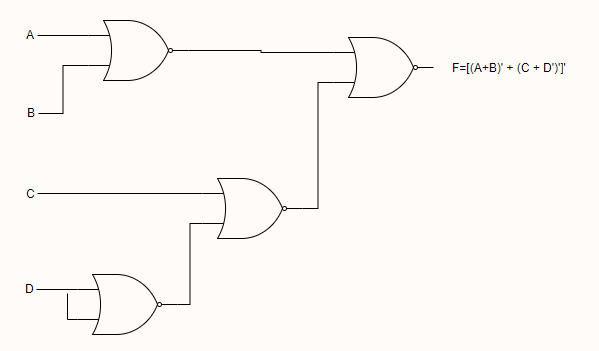
## Diagrama lógico NAND.



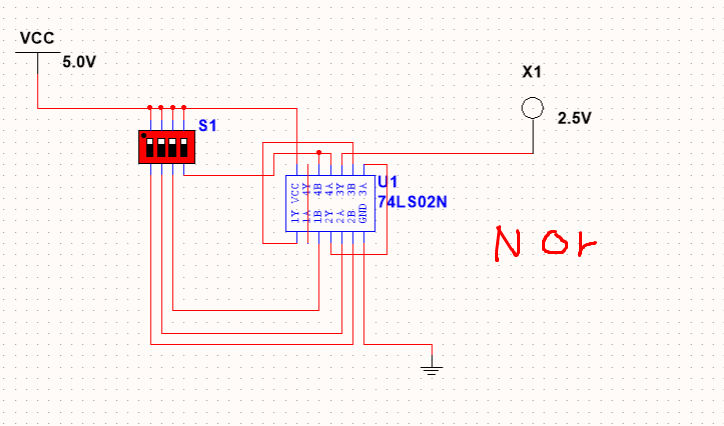
## Chipiograma NAND.



## Diagrama lógico NOR.



## Chipiograma NOR.

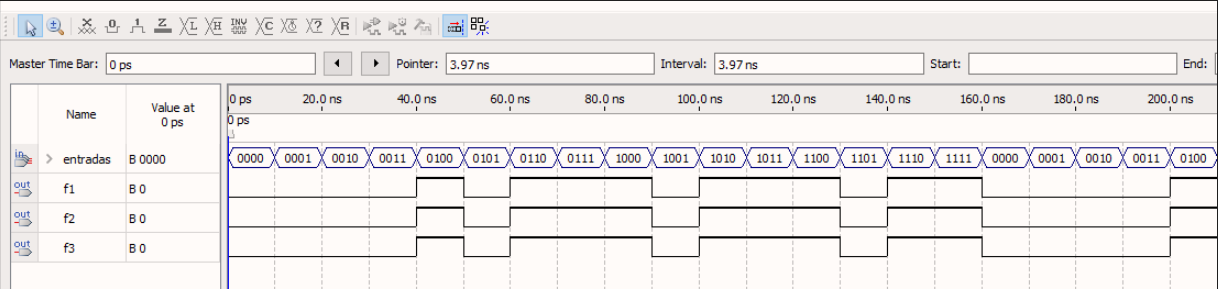


## Implementación del proyecto.

## Programa en VHDL y cómo se genera en VHDL

|  |
| --- |
| library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  use IEEE.STD\_LOGIC\_ARITH.ALL;  use IEEE.STD\_LOGIC\_UNSIGNED.ALL;  entity projecto\_1 is  port(  A : in std\_logic;  B : in std\_logic;  C : in std\_logic;  D : in std\_logic;  f1 : out std\_logic;  f2 : out std\_logic;  f3 : out std\_logic  );  end projecto\_1;  architecture BEHAVIORAL of projecto\_1 is  begin  f1 <= (A or B) and (C or not D);  f2 <= (A nor B) nor (c nor (D nor D));  f3 <= (((A nand A) nand (B nand B)) nand ((C nand C) nand D)) nand (((A nand A) nand (B nand B)) nand ((C nand C) nand D));    end BEHAVIORAL; |

## Simulación/implementación en VHDL



# Conclusión

Barrera Peña Víctor Miguel: Los circuitos lógicos salieron de acuerdo a lo planeado con la simulación, se podría decir que acuerdo a las entradas se obtienen los resultados esperados, se implementó en VHDL de acuerdo a las especificaciones profesores. La simulación la Max II se obtuvo pese a que no se sabía con exactitud era la correcta, pero se logró compilar.