

Diseño de celdas estándar en *Electric*

Nicolás David Arias Sosa **Código:** 261692 ndariass@unal.edu.co
 David Ricardo Martínez Hernández **Código:** 261931 drmartinezhe@unal.edu.co
 Oscar Alejandro Rojas Gallego **Código:** xxxxxx oarojasg@unal.edu.co
 Universidad Nacional de Colombia

Resumen—En la práctica de laboratorio que se describe a continuación se hizo el uso de la herramienta *Electric* para la implementación de circuitos básicos y celdas estándar usando una tecnología de proceso de 500 nm. De igual modo, se hizo la implementación de un layout para resolver un problema de aplicación.

Índice de términos—

I. INTRODUCCIÓN

A. Compuerta AND

La compuerta AND es una puerta lógica digital que implementa la conjunción lógica. Ésta entregará una salida **ALTA** (1), dependiendo de los valores de las entradas, siendo este caso, al recibir solo valores altos en la puerta AND. Si alguna de estas entradas no son **ALTAS**, entonces se mostrará un valor de salida **BAJA** (0). En otro sentido, la función de la compuerta AND efectivamente encuentra el mínimo entre dos dígitos binarios, así como la función OR encuentra al máximo. Se puede ver claramente que la salida **X** solamente es “1” (1 lógico, nivel alto) cuando la entrada **A** como la entrada **B** están en “1”. En otras palabras la salida **X** es igual a 1 cuando la entrada **A** y la entrada **B** son 1.

Esta situación se representa en álgebra booleana como:

$$X = A \cdot B \quad (1)$$

B. Compuerta NAND

La puerta NAND o compuerta NAND es una puerta lógica digital que implementa la conjunción lógica negada. Cuando todas sus entradas están en “1” o en **ALTA**, su salida está en 0 o en **BAJA**, mientras que cuando una sola de sus entradas o ambas están en 0 o en **BAJA**, su salida **X** va a estar en 1 o en **ALTA**.

Se puede ver claramente que la salida **X** solamente es “0” (0 lógico, nivel bajo) cuando la entrada **A** como la entrada **B** están en “1”. En otras palabras la salida **X** es igual a “0” cuando la entrada **A** y la entrada **B** son “1”.

Esta situación se representa en álgebra booleana como:

$$X = \overline{A \cdot B} = \overline{A} + \overline{B} \quad (2)$$

C. Flip-flop D

II. COMPUERTA AND DE 3 ENTRADAS

La compuerta AND de 3 entradas es la unión de 2 compuertas AND de 2 entradas, pero para este caso se utilizó la siguiente ecuación como las demostradas en clase.

Primero se utiliza la ecuación original para la parte N (ecu 3).

$$AND_{3IN} = A \cdot B \cdot C \quad (3)$$

Para la parte de los transistores tipo P, se utilizó la función negada (ecu 4).

$$\overline{AND_{3IN}} = \overline{A \cdot B \cdot C} = \overline{A} + \overline{B} + \overline{C} \quad (4)$$

Dando como resultado la Fig. 1, esta figura tiene en la función negada los transistores tipo P conectados a V_{CC} , y la función original con los transistores tipo N conectados a GND como en la Fig 1.

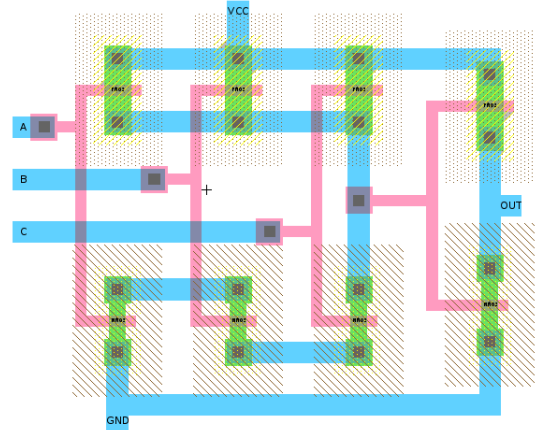


Fig. 1: Layout en electric de la compuerta AND de 3 entradas.

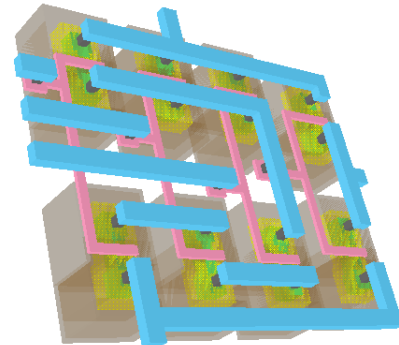


Fig. 2: Layout en electric en 3D de la compuerta AND de 3 entradas.

Como se puede observar en la Fig. 3 la salida de la compuerta AND con sus 3 respectivas entradas, en la Fig. 3 se puede observar en verde fosforescente.

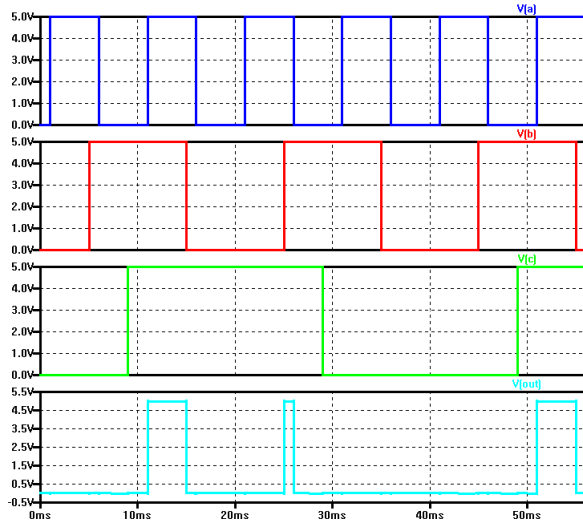


Fig. 3: Simulación en Spice de la AND 3 entradas implementado.

Como se puede observar la salida corresponde a la operación AND, solo cuando las 3 entradas se encuentren en 1 lógico la salida también será un 1 lógico, de lo contrario la salida estará en 0 lógico.

III. DISEÑO FLIP-FLOP D

En la fig. 4 se muestra el diagrama esquemático del flip-flop D a implementar, donde el bloque *C* representa una compuerta de transmisión. Para la elaboración del layout se usó el inversor desarrollado en la práctica pasada, de modo que fue posible insertar varias instancias del módulo.

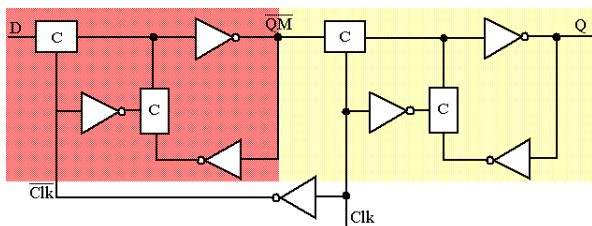


Fig. 4: Esquemático del flip-flop D a implementar. [1]

En las fig. 5 y 6 se muestran los layout en *Electric* del flip-flop y de la compuerta de transmisión, correspondientemente.

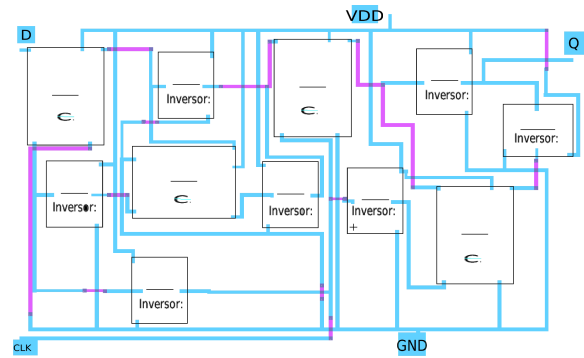


Fig. 5: Layout en electric del flip-flop D

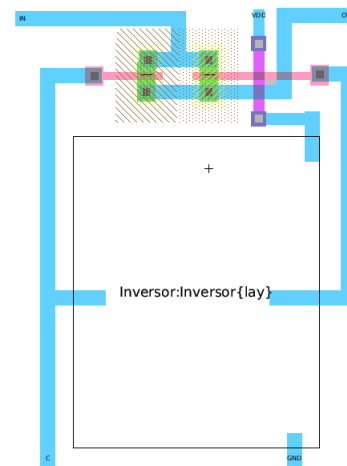


Fig. 6: Layout en *Electric* de la compuerta de transmisión

La salida *Q* del flip-flop toma el valor de la entrada *D* únicamente en los flancos de subida del reloj *Clk*. Por tanto, el flip-flop funciona como un elemento de almacenamiento. La implementación consiste en la unión de dos latches sensibles por nivel: un *master* (color rosa en la fig. 4) cuya salida *Q_m* toma el valor de *D* para niveles bajos del reloj; y un *slave* (color amarillo en la fig. 4) que toma el valor de *Q_m* para niveles altos del reloj. De ahí entonces que la salida solo sigue a la entrada en los flancos de subida del reloj. Este comportamiento se ilustra en la fig. 7.

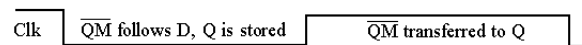


Fig. 7: Descripción del comportamiento de los latches de acuerdo al nivel del reloj *Clk*.

En la fig. 8 se muestra la simulación del flip-flop obtenida en Spice. Para $t = 15, 25, 45$ ms, que corresponden a flancos de subida del reloj, se tiene que la señal de salida actualiza su valor acorde a la señal de entrada.

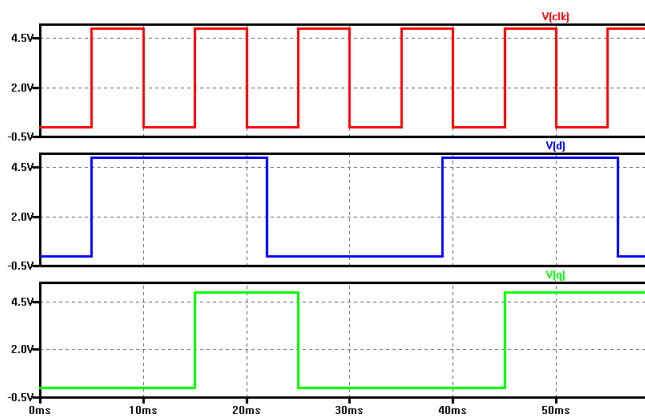


Fig. 8: Simulación en *Spice* del flip-flop implementado.

A. Planteamiento

IV. ANÁLISIS DE RESULTADOS Y CONCLUSIONES

A. Compuerta *AND* de 3 entradas

B. Diseño *flip-flop D*

C. Planteamiento

REFERENCIAS

- [1] The University of New Mexico. '*Pass Gates Applications: Latches and Registers*'. Disponible en http://www.ece.unm.edu/~jimp/vlsi/slides/c1_basics.html
- [2] Sedra, Adel S. & Smith, Kenneth C. "'*Circuitos Microelectrónicos*'". Oxford University Press, Cuarta Edición, 1999.