

# Implementación de un contador para un estacionamiento - 2019

## Práctica de laboratorio N°2 Electrónica Digital

Michel Gartner  
michelgartner8@gmail.com

8 de septiembre de 2019

### 1. Especificación de requerimientos del sistema

Implementación de un dispositivo para contar la cantidad autos que ingresaron y permanecen en un estacionamiento en una placa Nexys3 Spartan6 XC6LX16-CS324.

Se cuenta con dos sensores en serie separados una distancia menor al largo del auto. Cuando un auto se ubica a la altura de uno de los sensores, se genera un bit 1, en caso contrario es un 0.

Luego, el sistema funciona al reconocer dos eventos específicos, el ingreso de un auto y el egreso de un auto. El ingreso de un auto ocurre cuando ocurre la secuencia  $(0, 0) \rightarrow (1, 0) \rightarrow (1, 1) \rightarrow (0, 1) \rightarrow (0, 0)$  mientras que el egreso del auto ocurre cuando ocurre la secuencia  $(0, 0) \rightarrow (0, 1) \rightarrow (1, 1) \rightarrow (1, 0) \rightarrow (0, 0)$ .

Como criterio de este diseño se considera como error a cualquier transición de estados que no este contenida en las secuencias anteriores. En estos casos, el sistema vuelve al estado  $(0, 0)$ .

Para la implementación en la placa los sensores se mapean a los switches s0 y s1 de la placa. El resultado de la cuenta de eventos se muestra en los 4 módulos siete segmentos en representación hexadecimal.

El reset del circuito se mapea en el pb2, mientras que el clock es el clock de la placa.

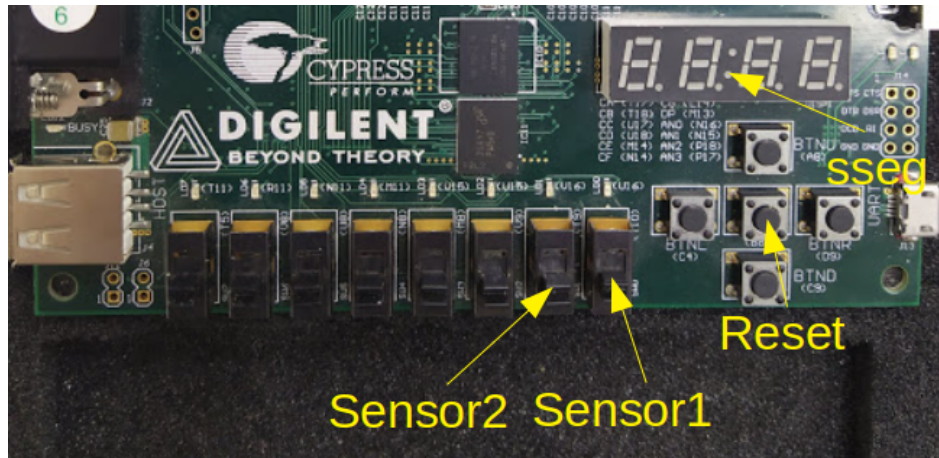


Figura 1: Ubicación de las partes más significativas de la placa Nexys3 para la realización del contador de estacionamiento.

En la figura 1 se muestra la ubicación de los switches, los push buttons, los siete segmentos y el led mencionados anteriormente.

## 2. Diseño del módulo TOP

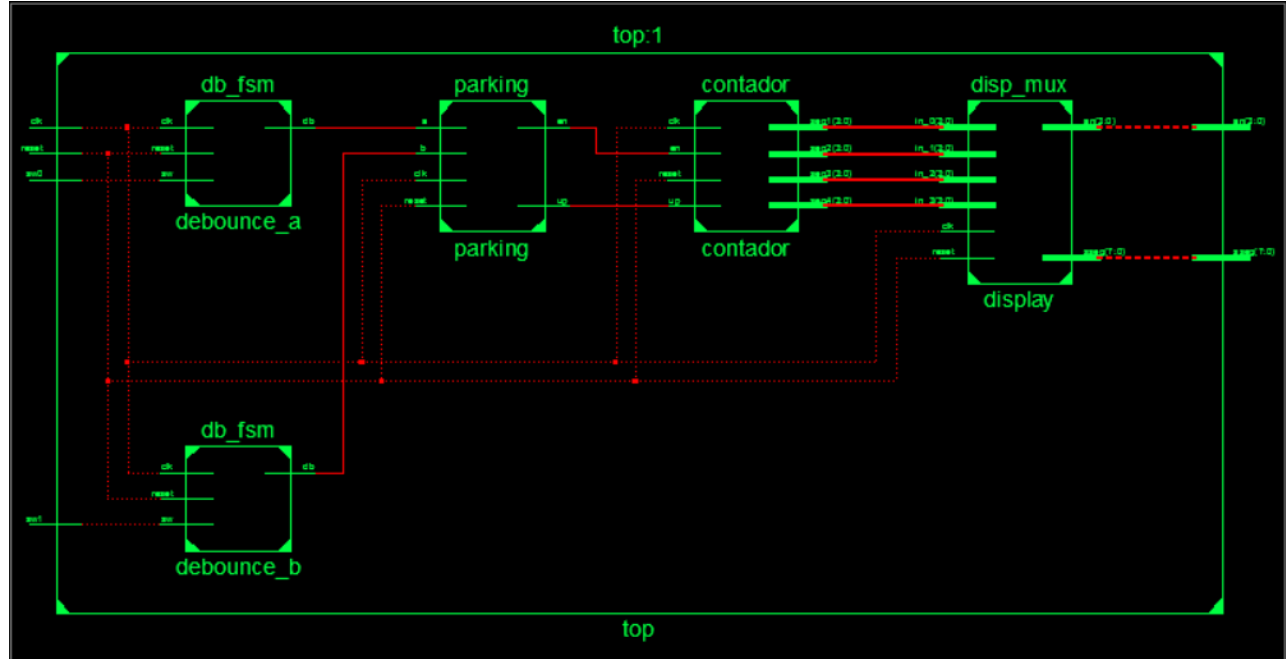


Figura 2: Diagrama de bloques de la lógica del módulo 'top'.

En la figura 2 se muestra esquemáticamente en diagrama de bloques la lógica implementada en el módulo 'top' del estacionamiento.

Dado que los bits de entrada 'a' y 'b' están mapeados a los switches, estos pueden producir una señal inestable durante un breve intervalo de tiempo. Por ello, cada uno de estos valores ingresan al módulo de debounce 'db fsm' que tiene como salida la parte estable de sus entrada. Estas salidas ingresan al módulo 'parking' para identificar los eventos de interés. En caso que se produzca un evento de interés, la salida 'en' es '1'. Si se produce un ingreso, la señal de salida 'up' es '1', mientras que para un egreso la señal 'up' es 0.

Las salidas 'up' y 'en' ingresan al módulo 'contador'. Este módulo solo cambia su valor cuando 'en' es 1. En esta situación cuando 'up' es 1 se suma mientras que cuando 'up' es 0 se resta.

La salida de del módulo contador son los 4 números hexadecimal que se corresponden con cada uno de los 4 display 7 segmentos. Por ello, estos valores ingresan al módulo 'disp mux' que muestra sus cuatro entradas en los 4 7 segmentos.

### 3. Sub-módulos

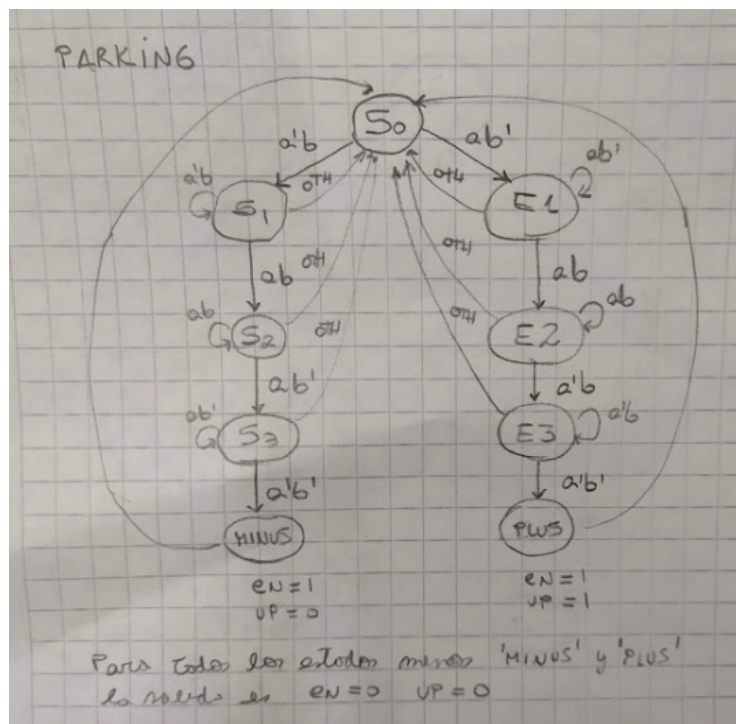


Figura 3: Diagrama de estados del módulo 'parking'

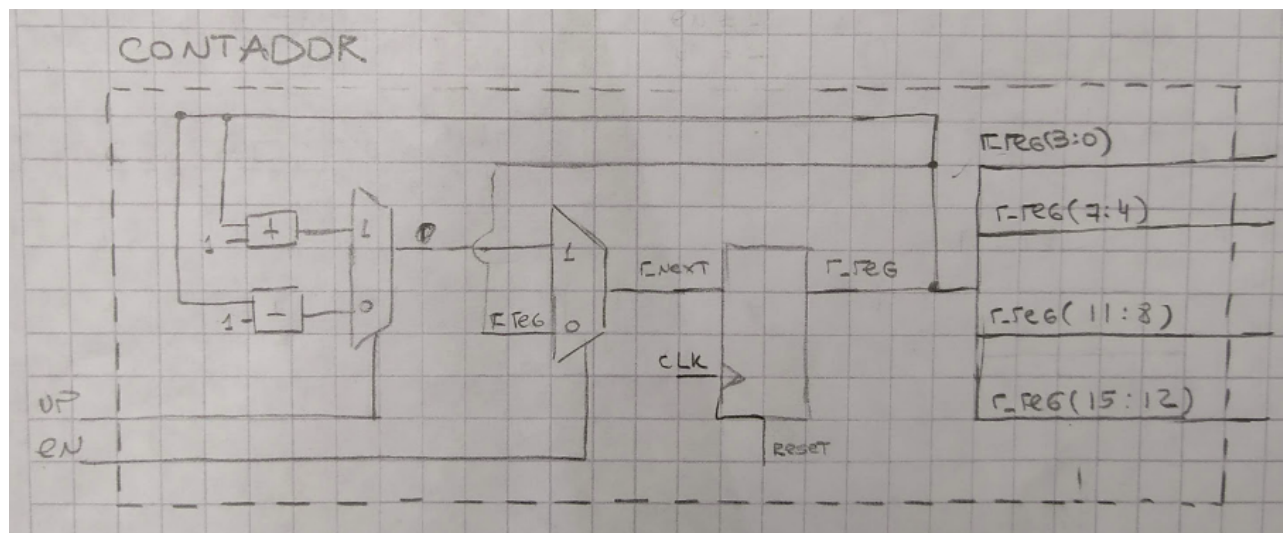


Figura 4: Diagrama de bloques de la lógica del módulo 'contador'.

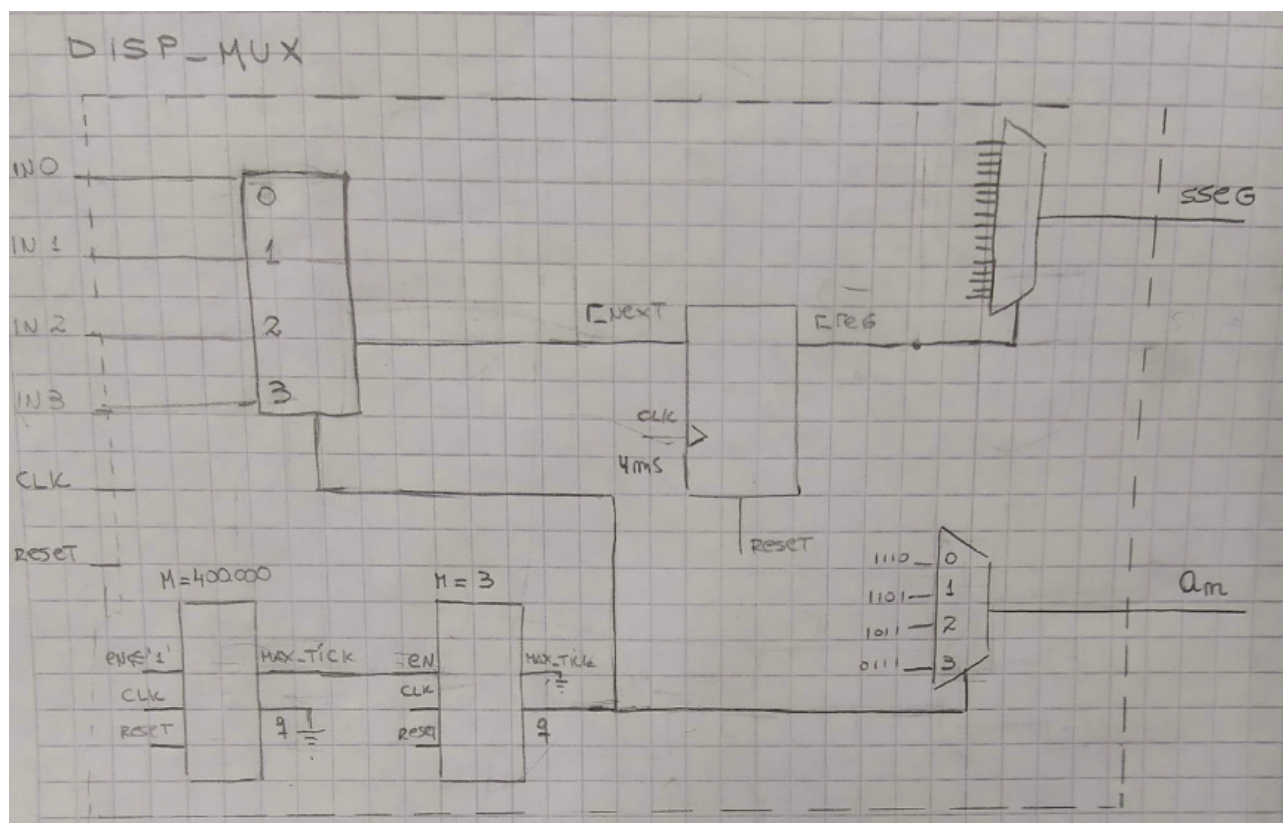


Figura 5: Diagrama de bloques de la lógica del módulo 'disp mux'.

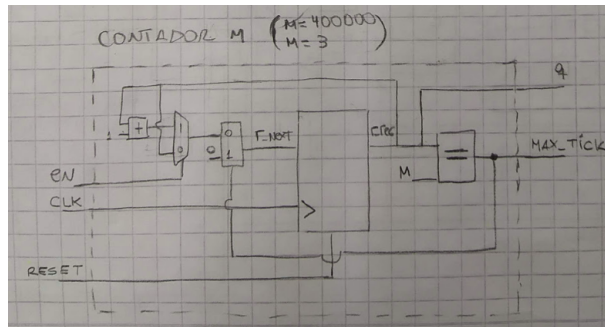


Figura 6: Diagrama de bloques de la lógica del sub módulo 'cont M' utilizado en 'disp mux'.

En la figura 3 se muestra el diagrama de estados del módulo 'parking'. En la figura 4, 5 se muestran los diagramas de bloques de la lógica de los módulos 'contador' y 'disp mux' respectivamente. Por último, en la figura 6 se muestra el submódulo 'cont M' utilizado en el diagrama de 'disp mux'.

## 4. Conclusiones

Se implementó en una placa Nexys3 un contador de autos que ingresan a un estacionamiento. Se codificó en lenguaje VHDL utilizando compuertas lógicas y bloques combinacionales. Se realizaron los test-bench correspondientes para los módulos más relevantes y se verificó en la placa.