

# ELECTRÓNICA DIGITAL - LABORATORIO III

## Implementación de Contador Binario Controlado por Máquina de Estados Finita

MATÍAS DANIEL ROQUETA

Centro Atómico Bariloche y Instituto Balseiro, Comisión Nacional de Energía Atómica

### Introducción

Se requiere implementar un circuito digital que cuente el número de autos presentes en un estacionamiento. Para esto se tienen dos fotosensores ubicados a la entrada del estacionamiento, arreglados según la figura 1.

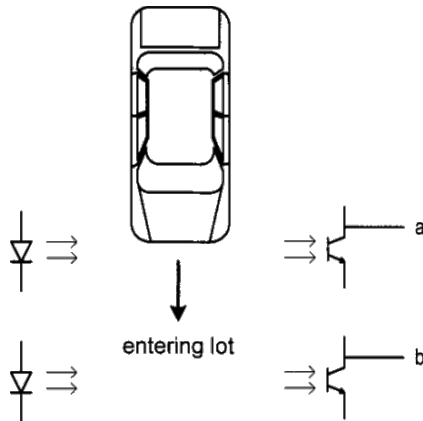


Figura 1: En un estado de espera, ambas señales están en LOW. Cuando el auto se encuentra en frente del sensor A, la correspondiente señal pasa al estado HIGH. Asimismo el sensor B.

Para implementar el circuito digital se implementa una máquina de estados finita (FSM) para detectar eventos, y el número de autos se registra con un contador universal síncrono (`sync_count`). Se implementa un reset asíncronico, y una opción de carga paralela para fijar manualmente la cantidad de autos presentes en el estacionamiento.

### Arquitectura

La primer etapa de diseño consiste en implementar la máquina de estados finita capaz de monitorear los eventos de entrada y salida de autos.

En primer instancia se opta por una máquina tipo Mealy por su menor cantidad de estados necesarios. La máquina diseñada cuenta con tres estados

- **wait**: La máquina está en estado de espera.
- **entr**: La máquina comenzó a detectar un ingreso.
- **exit**: La máquina comenzó a detectar un egreso.

Y con dos salidas

- **entr\_sig**: La máquina informa el ingreso de un auto.
- **exit\_sig**: La máquina informa el egreso de un auto.

Los estados y salidas responden a las entradas A, B según la siguiente tabla de transiciones <sup>1</sup>

Estado	AB							
	00		10		01		11	
wait	wait	—	entr	—	exit	—	wait	—
entr	wait	—	entr	—	wait	—	wait	entr_sig
exit	wait	—	wait	—	exit	—	wait	exit_sig

Tabla 1: Transiciones que definen a la FSM de control.

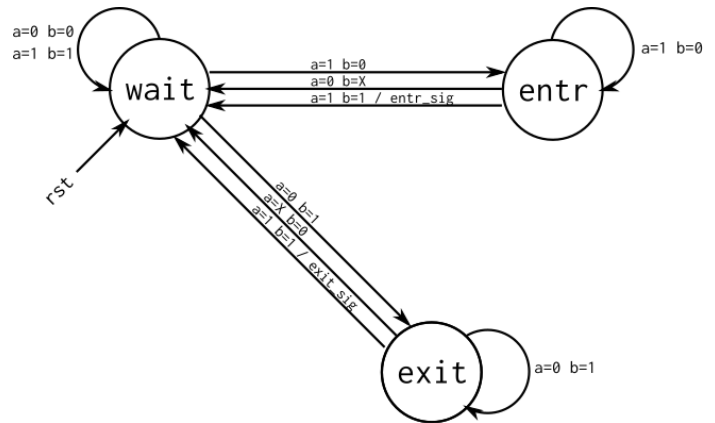


Figura 2: Diagrama de estados correspondiente a la tabla 1.

La máquina de estados reacciona al ingreso de un auto de la siguiente forma:

1. **wait**: Un auto acciona el sensor A/AB=10, transición **wait**→**entr** sin señal de salida.
2. **entr**: El auto acciona el sensor B/AB=11, transición **entr**→**wait** con señal **entr\_sig** informando ingreso.
3. **wait**: El auto libera al sensor A/AB=01, transición **wait**→**exit** sin señal de salida.
4. **exit**: El auto libera al sensor B/AB=00, transición **exit**→**wait** sin señal de salida.

Por la simetría del problema, el procedimiento es equivalente en el caso de la salida de un auto del estacionamiento.

La siguiente etapa consiste en tomar las señales **entr\_sig** y **exit\_sig** e implementar un circuito que controle al contador universal síncrono en función de ellas. Esta etapa también implementará la funcionalidad de carga paralela.

Partimos de la tabla de verdad del contador universal para extenderla con las señales de control.

<sup>1</sup>Esta no es la forma canónica de describir la tabla de transiciones de una máquina Mealy, es una versión reducida por el carácter de las salidas. En la columna de salidas, una salida está en HIGH si está explicitada en la transición, de lo contrario está en LOW.



**wait** : Estado de espera.

**entr\_x** : Estado de ingreso incompleto.

**entr\_v** : Estado previo a validación de ingreso.

**exit\_x** : Estado de egreso incompleto.

**exit\_v** : Estado previo a validación de egreso.

La máquina de estados Mealy propuesta queda totalmente definida por la tabla 3.

Estado	AB							
	00		10		01		11	
wait	wait	—	entr_x	—	exit_x	—	wait	—
entr_x	wait	—	entr_x	—	entr_v	—	entr_x	—
entr_v	wait	entr_s	entr_v	—	entr_v	—	entr_x	—
exit_x	wait	—	exit_v	—	exit_x	—	exit_x	—
exit_v	wait	exit_s	exit_v	—	exit_v	—	exit_x	—

Tabla 3: Transiciones que definen a la FSM alternativa de control.

Para evaluar el funcionamiento de la arquitectura alternativa se define la señal de control según

```
ctrl = q.now Mux{wait_q : ZZ, entr_x : 00, entr_v : 01,
                  exit_x : 01, exit_v : 11}
```

Y se somete a un test equivalente al del módulo FSM anterior, resulta la simulación presentada en la figura 6. Si ahora se reemplaza la instancia del módulo FSM en el top level por su versión alternativa y se ejecuta el mismo test sobre el mismo se obtiene la simulación presentada en la figura 7.

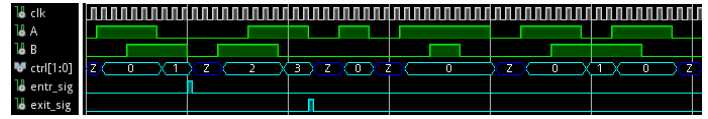


Figura 6: Resultado de las pruebas al módulo FSM alternativo.

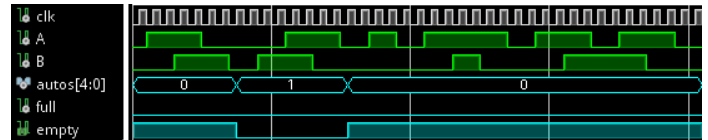


Figura 7: Resultado de las pruebas al módulo top level instanciando la arquitectura alternativa de FSM.

Se verifica que con la arquitectura alternativa, el dispositivo es ahora a prueba de errores al costo de un circuito digital más complejo. Ambas arquitecturas se incluyen en el archivo .vhd adjunto.