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.

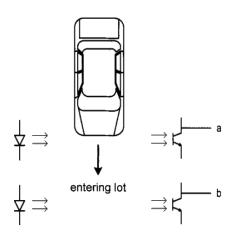


Figura 1: En un estado de espera, ambas señales están en LOW. Cuando el auto se encuentra en frente del sensor A, la correspondiente señal pasa al estado HIGH. Asimismo el sensor B.

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 sincrónico (sync_count). Se implementa un reset asincrónico, 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 $^{\rm 1}$

Estado	AB							
	00		10		01		11	
wait	wait	_	entr	_	exit	_	wait	_
entr	wait	-	entr	_	wait	_	wait	
exit	wait	-	wait	_	exit	_	wait	exit_sig

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

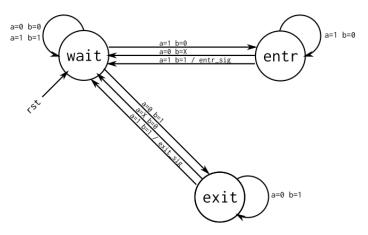


Figura 2: Diagrama de estados correspondiente a la tabla 1.

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

- wait: Un auto acciona el sensor A/AB=10, transición wait→entr sin señal de salida.
- entr: El auto acciona el sensor B/AB=11, transición entr→wait con señal entr_sig informando ingreso.
- wait: El auto libera al sensor A/AB=01, transición wait→exit sin señal de salida.
- 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 saida 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 sincrónico 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.

¹Esta 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.

entr_sig	exit_sig	en	up	q*
0	0	0	_	q
0	1	1	0	q-1
1	0	1	1	q+1
1	1	_	_	_

Tabla 2: Tabla de verdad del módulo sync_count extendida con las señales de control que accionan el correspondiente resultado. Notar que los resultados correspondientes a la entrada 11 no importa ya que esta combinación es imposible.

Se plantean las asignaciones compatibles con la tabla 2

Con esta expresión lógica se instancia el módulo top_level, incluyendo las señales de reloj, reset asincrónico, y la funcionalidad de carga paralela.

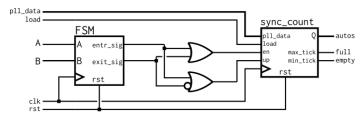


Figura 3: Esquemática del top level instanciado en VHDL.

Este diseño modular permite modificar la implementación del módulo sync_count o de la FSM mientras estas mantengan su interfaz.

Tests

Hay dos tests importantes que realizar, el test de la FSM, y el test del top level, que necesitan ser realizados en ese orden.

Para evaluar el funcionamiento de la FSM se agrega una señal de salida adicional que permite observar el estado actual en un determinado momento, definida por 2

$$\mathtt{ctrl} = \mathtt{q}\mathtt{.now}\,\mathrm{Mux}\,\{\mathtt{wait}\mathtt{_q}:00,\,\,\mathtt{entr}\mathtt{_q}:01,\,\,\mathtt{exit}\mathtt{_q}:10\}$$

Monitoreando la señal ctrl y las señales de salida se puede garantizar que la máquina de estados se comporte según las transiciones de la tabla 1

Los casos de prueba son de ingreso/egreso exitoso

Ingreso: AB =
$$00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00$$
.

Egreso :
$$AB = 00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00$$
.

Seguidos de posible casos de error, que evalúan que sucede en caso de que un auto inicie un proceso de ingreso, lo interrumpa en algún momento, y lo revierta.

Error 1 : $AB = 00 \rightarrow \underline{10} \rightarrow 00$.

Error 2: AB = $00 \rightarrow 10 \rightarrow \underline{11} \rightarrow 10 \rightarrow 00$.

Error 3: $AB = 00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00$.



Figura 4: Resultado de los casos de prueba al módulo FSM.

La figura 4 adelanta que la máquina de estados transiciona tal como esperado e informa ingresos y egresos correctamente. En caso de error 1 no informa nada, en caso de error 3 informa un ingreso inmediatamente contrarrestado por un egreso, pero el caso de error 2 representa una vulnerabilidad del diseño.

El test del módulo top level instancia un contador de 5 bits y se puede dividir en dos etapas, de las cuales la primera es una prueba de contabilización. Esta prueba evalúa la respuesta del contador ante las mismas entradas implementadas en el test de la FSM.

La segunda etapa consiste en validad el funcionamiento de la carga paralela y la salida full.

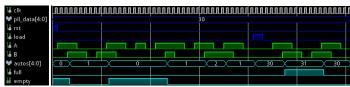


Figura 5: Resultado de las pruebas al módulo top level.

El resultado de la prueba al top level era el esperado luego de las pruebas al módulo FSM. Efectúa la cuenta ascendente y descendente correctamente en su correcto uso, pero mantiene una vulnerabilidad al caso de error 2 que desfasa la cuenta de ahí en adelante. La simulación eleva un warning de cuenta incorrecta a partir de ese momento.

Discusión

Que sea un problema el caso de error 2 depende totalmente del uso del dispositivo. Si nunca se va a presentar una situación en la que un auto accione el sensor y luego retroceda (por ejemplo, si existe una barrera), entonces no tiene justificación diseñar una máquina de estados robusta ante esos errores.

Corregir ese caso de error implica necesariamente una mayor cantidad de estados, un análisis provisorio sugiere que se necesitan 5 estados, requiriendo extender una FSM de 3 bits en lugar de 2. Más la implementación de la lógica de estado siguiente esto incrementa significativamente la complejidad del diseño y solo se justifica si es verdaderamente necesario.

Sin embargo, el diseño modular implica que de ser necesario, una vez diseñada la arquitectura de tal máquina remplazarla en la implementación top level es una tarea simple.

Se propone, por ejemplo, una implementación de una máquina de estados Mealy que corrige el error mencionado pero requiere 5 estados, los cuales son

²En la implementación VHDL los estados wait, entr, exit se definen con wait_q, entr_q, exit_q por ser wait y exit palabras reservadas.

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								
Lstado	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

$$\begin{split} \mathtt{ctrl} = \mathtt{q_now} \, \mathrm{Mux} \{ \mathtt{wait_q} : ZZ, \, \mathtt{entr_x} : 00, \, \, \mathtt{entr_v} : 01, \\ \mathtt{exit_x} : 01, \, \, \mathtt{exit_v} : 11 \} \end{split}$$

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.

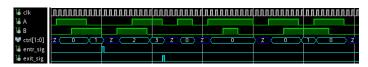


Figura 6: Resultado de las pruebas al módulo FSM alternativo.

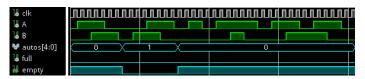


Figura 7: Resultado de las pruebas al módulo top level instanciando la arquitectura alternativa de FSM.

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.