

Diseño de Controlador de Semáforo

Autor: Díaz Gerardo Agustín

Carrera: Ingeniería Electrónica

Materia: Electrónica II

Año: 2025

Resumen

Se diseñó e implementó en VHDL un controlador de semáforo digital para una intersección, empleando flip-flops D y temporizadores sincrónicos de 12 MHz. Se abordaron dos metodologías de diseño: una máquina de estado finito única y una MEF factorizada en módulos de control y temporización. Se añadieron modos de emergencia y de cruce peatonal, verificando el funcionamiento mediante simulación y banco de pruebas en la placa EDU-CIAA-FPGA.

Introducción

El presente trabajo tiene como objetivo diseñar e implementar un controlador digital de semáforo empleando una MEF codificada en lenguaje VHDL, considerando las señales de emergencia y de cruce peatonal, además del ciclo normal de funcionamiento. El sistema se basa en una arquitectura sincrónica, gobernada por un reloj de 12 MHz y un temporizador derivado del mismo, que permite generar los intervalos temporales de 60 s, 50 s y 10 s requeridos para cada fase del semáforo.

Materiales y Métodos

El diseño adoptado se estructuró en cuatro estados principales: Verde A, Amarillo A, Verde B y Amarillo B, que corresponden a las dos direcciones del cruce (A y B).

Cada dirección transita por un ciclo Verde → Amarillo → Rojo, garantizando la exclusión mutua: cuando una dirección está en Verde o Amarillo, la otra permanece necesariamente en Rojo. De esta forma se asegura la secuencia de tránsito sin conflictos.

Asimismo, se incorporaron estados adicionales de emergencia y de cruce peatonal:

Ante la activación de una señal de emergencia, el controlador interrumpe la secuencia normal y otorga prioridad inmediata a la dirección correspondiente, manteniéndola en verde hasta que se libera la condición.

Cuando se acciona un pulsador peatonal, se extiende el tiempo del verde de la dirección seleccionada, duplicando su duración o activando un tiempo especial en el que el peatón puede cruzar con seguridad, mientras todos los autos están en rojo.

Diseño como FSM factorizada

En esta versión del diseño se empleó una máquina de estado finito factorizada, lo que significa que el sistema completo se divide en dos módulos principales y complementarios:

- El módulo de Control
- El módulo de Temporización.

Módulo de Control

El módulo de control contiene la lógica que determina qué luces deben encenderse y cuándo cambiar de estado.

Se basa en una MEF de tipo Moore, donde cada estado representa una situación particular del semáforo, como, por ejemplo:

Verde A, Amarillo A, Verde B, Amarillo B, más los estados especiales de Emergencia y Cruce Peatonal.

Módulo de Temporización

El temporizador es un bloque independiente que genera las señales de tiempo necesarias (tick 10s, tick 50s, tick 60s) a partir del reloj de 12 MHz.

Internamente utiliza un prescaler para reducir la frecuencia del reloj a 1 Hz y un contador para medir segundos.

Cada vez que se cumple el tiempo configurado, el temporizador emite un pulso de 1 ciclo que indica a la MEF de control que debe avanzar al siguiente estado.

Resultados

En la Figura 1 se muestra el diagrama de estados del controlador de semáforo, donde se representan los cuatro estados principales Verde A, Amarillo A, Verde B, Amarillo B y los estados especiales de emergencia y cruce peatonal.

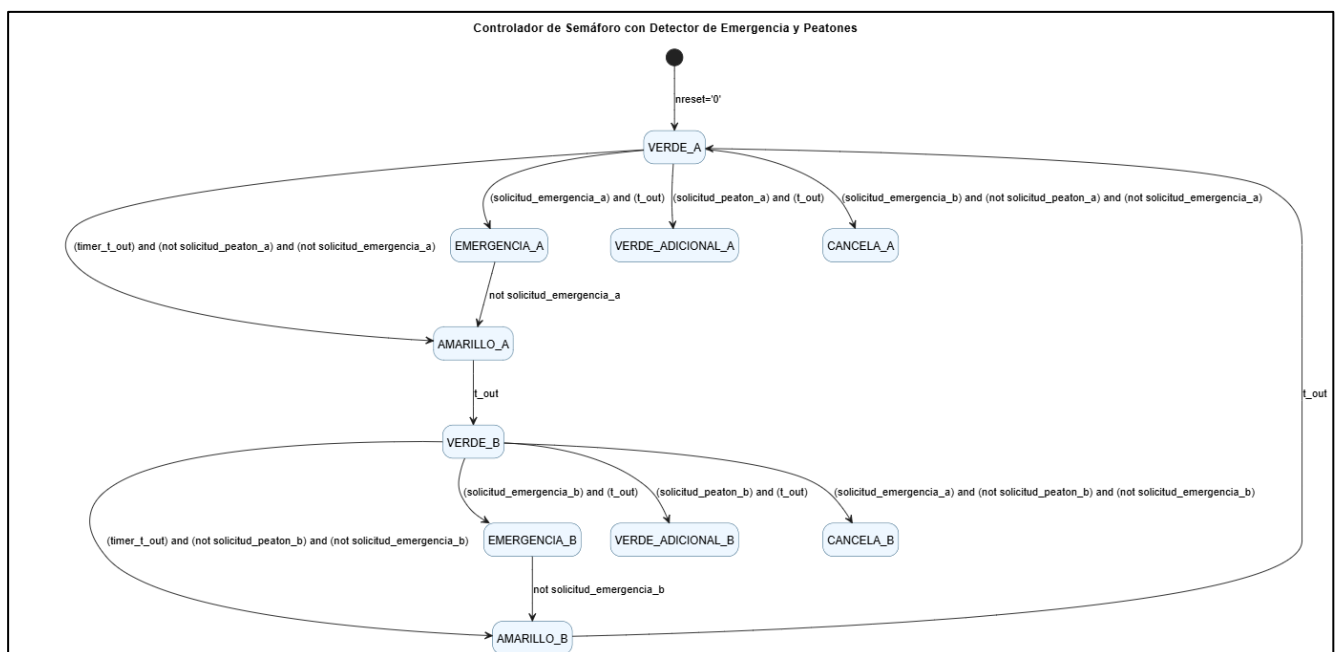


Figura 1: Diagrama de estados del controlador de semáforos (generado con PlantUML)

- **Control y tiempos del sistema de semáforos**

El controlador alterna VERDE_A → AMARILLO_A → VERDE_B → AMARILLO_B → ...

Cada estado verde dura T_VERDE y el amarillo T_AMARILLO. La conmutación se habilita cuando el temporizador indica fin (t_out='1').

- **Atención a peatones (prioridad menor que emergencia)**

Cada botón (solicitud_peaton_X) se guarda en m_peaton_X.

Al entrar a VERDE_A (o VERDE_B), si hay m_peaton_X='1', el sistema va hacia VERDE_ADICIONAL_X donde se extiende el tiempo del peatón y mantiene verde X, se asigna carga_timer := T_PEATON, y la calle está en rojo para vehículos de la otra dirección. Al terminar, emite confirmacion_peaton_X='1' para limpiar m_peaton_X.

- **Emergencia (prioridad más alta)**

Si llega solicitud_emergencia_A='1' mientras estaba VERDE_B (o viceversa), el sistema fuerza transición. Pasa por un estado de transición CANCELA_B (pone amarillo la calle B que estaba

verde) por T_AMARILLO, luego entra a EMERGENCIA_A (mantiene verde a A, rojo a B) mientras dure la emergencia, cuando solicitud_emergencia_A vuelve a '0', se libera y vuelve al ciclo normal.

- **Temporizador**

Un temporizador genera t_out al agotarse la cuenta. El controlador carga carga_timer según el estado o según peatón/emergencia.

- **Reset**

nreset='0' coloca el sistema en VERDE_A, apaga peatones, y limpia las memorias (m_peaton_X) y detectores de flanco.

Entradas y salidas

La Tabla 1 resume todas las señales utilizadas en el diseño del controlador de semáforo, especificando su dirección (entrada o salida), tipo de dato y propósito dentro del sistema.

Señal	Dirección	Tipo	Descripción
clk	in	std_logic	Señal de reloj principal del sistema. Sincroniza la máquina de estados y el temporizador. Cada flanco ascendente representa un ciclo de operación.
nreset	in	std_logic	Reset activo en '0'. Al activarse, reinicia el sistema colocando el semáforo en su estado inicial (por defecto, VERDE_A) y limpia todas las señales internas.
solicitud_peaton_a	in	std_logic	Entrada del botón peatonal de la calle A. Al presionarlo, se solicita tiempo de cruce para peatones de ese lado.
solicitud_peaton_b	in	std_logic	Entrada del botón peatonal de la calle B. Cumple la misma función que la anterior, pero para la vía opuesta.
solicitud_emergencia_a	in	std_logic	Señal que indica la presencia de un vehículo de emergencia en la calle A. Activa el modo de emergencia para darle paso.
solicitud_emergencia_b	in	std_logic	Señal equivalente para la calle B. Tiene prioridad sobre el ciclo normal y sobre los peatones.
transito_a	out	std_logic_vector	Control de las luces vehiculares del semáforo A
transito_b	out	std_logic_vector	Control de las luces vehiculares del semáforo B
confirmacion_peaton_a	out	std_logic	Indica pedido de peatón A atendido
confirmacion_peaton_b	out	std_logic	Indica pedido de peatón B atendido

Tabla 1: *Señales de entrada y salida*

Estados del controlador de semáforo

En la Tabla 2 se presenta el mapeo entre los estados de la máquina de control y las salidas correspondientes, junto con los tiempos asignados a cada fase del semáforo.

Estado	Vehicular A	Vehicular B	Peatón A	Peatón B	carga_timer	hab_timer
VERDE_A	Verde ("01")	Rojo ("10")	1	0	T_VERDE	1
VERDE_ADICIONAL_A	Verde ("01")	Rojo ("10")	1	0	T_PEATON	1
AMARILLO_A	Amarillo("11")	Rojo ("10")	0	0	T_AMARILLO	1
VERDE_B	Rojo ("10")	Verde("00")	0	1	T_VERDE	1
VERDE_ADICIONAL_B	Rojo ("10")	Verde("01")	0	1	T_PEATON	1
AMARILLO_B	Rojo ("10")	Amarillo("11")	0	0	T_AMARILLO	1
EMERGENCIA_A	Amarillo A ("01")	Rojo B ("10")	1	0	0 (No cuenta, se desactiva mientras hay emergencia)	0
EMERGENCIA_B	Verde A ("00")	Rojo B ("10")	0	1	0 (No cuenta, se desactiva mientras hay emergencia)	0
CANCELA_A	-	-	0	0	0	0
CANCELA_B	-	-	0	0	0	0

Tabla 2: *Mapeo de estados*

Conclusiones

El desarrollo de la Actividad 6 permitió aplicar de forma integrada los conceptos fundamentales de la lógica secuencial, el uso de flip-flops D y el diseño de máquinas de estado finito (FSM) en un sistema de control real: un semáforo digital.

A partir del reloj de 12 MHz se implementó un temporizador síncrono que definió los intervalos de 60, 50 y 10 segundos, garantizando precisión temporal y funcionamiento estable.

En cuanto al método de diseño, se comprobó que la factorización de la MEF en dos módulos uno de control lógico y otro de temporización simplifica la síntesis y mejora la estructuración del sistema.

Esta estructura permite reutilizar el temporizador en otros proyectos y facilita la corrección de errores, en comparación con una FSM monolítica que integra todo el control en un solo bloque.

Referencias

- Harris, S. L., & Harris, D. (2019). Digital design and computer architecture: RISC-V edition. Morgan Kaufmann.