

---

## Laboratorio 3: Lógica Secuencial y Controladores

---

Fecha de asignación: 4 abril 2025  
Grupos: 3 personas

Fecha de entrega: 7 mayo 2025  
Profesores: Jeferson González, Luis Barboza

---

### 1. Introducción

Un circuito secuencial es aquel en que las salidas dependen tanto de las combinaciones de la entradas y del tiempo. Esta característica hace que los sistemas secuenciales sean poderosos en tareas que involucran tiempo, a la vez que representan un mayor grado de complejidad que los combinacionales. En sistemas digitales complejos, como los microprocesadores y controladores de dispositivos externos, los circuitos de lógica secuencial juegan un papel fundamental. La lógica secuencial, al tomar en cuenta el tiempo, ha sido utilizada además para la creación de controladores de diferentes dispositivos en los computadores modernos. Cada componente de hardware que se anexa a un computador tiene su protocolo para configuración y uso. Uno de los dispositivos más comunes anexados a un computador es el dispositivo de visualización, como es el caso de un monitor. Existen diferentes tecnologías y protocolos de configuración de monitores. Para este laboratorio se trabajará con monitores que trabajan con una interfaz VGA (Video Graphics Array) en una resolución de 640x480 píxeles.

En este laboratorio el estudiante aplicará los conceptos de lógica secuencial en el diseño de circuitos digitales lógicos con un lenguaje de descripción de hardware. Además, se abordarán asuntos de tiempo relacionados con sincronización, así como rebotes en señales digitales por elementos mecánicos.

### 2. Investigación

Para el desarrollo de este laboratorio se deben responder las siguientes preguntas.

1. Investigue sobre el funcionamiento de máquinas de estado finitos. Explique la diferencia entre una máquina de Moore y una de Mealy y muestre la diferencia por medio de diagramas de estados y señales.
2. Explique los conceptos de *setup time* y *hold time*. ¿Qué importancia tienen en el diseño de sistemas digitales?
3. Investigue sobre el efecto de rebote en señales digitales provenientes de elementos mecánicos (interruptores, por ejemplo). Muestre al menos dos formas de solucionar el efecto de rebote, por medio de circuitos digitales.

4. Investigue sobre las señales involucradas en la sincronización de una interfaz VGA.
5. Muestre un diagrama de tiempos de las señales de sincronización de VGA para una resolución de 640x480 pixeles.
6. Para la resolución del punto anterior, calcule matemáticamente la frecuencia aproximada de las señales de sincronización vertical y horizontal.
7. Proponga un diagrama de bloques que implemente el controlador de VGA. Tenga en cuenta que este será parte de su diseño final, utilizando un modelado de estructura.

### 3. Ejercicio Práctico

A continuación se presenta un ejercicio práctico, el cual debe resolver de manera completa.

#### 3.1. Problema Planteado

Para el problema se deberán diseñar dos módulos: un controlador de VGA, para una resolución de 640x480 pixeles, así como un controlador (máquina de estados finita) que permita implementar una aplicación, utilizando el controlador de VGA.

##### 3.1.1. Controlador VGA

Para el diseño del controlador VGA debe realizar un modelo de estructura, en SystemVerilog, que incluya los subcomponentes del mismo. Tome en cuenta la naturaleza de las señales, así como el diagrama de tiempos investigado en la sección anterior. El controlador deberá manejar las señales de sincronización, así como las señales de color por pixel (8 bits por canal, por pixel), que serán generadas tomando en cuenta el controlador principal de la aplicación. El controlador deberá diseñarse para una resolución de 640x480 pixeles.

##### 3.1.2. Aplicación *Connect 4* y controlador principal

La aplicación a implementar será el juego Connect 4 (4 en línea). A continuación se muestran las especificaciones de la aplicación:

1. El tamaño del tablero es 6x7.
2. La partida será entre dos jugadores.
3. El jugador 1 realizará las jugadas por medio de los botones/switches de la FPGA.
4. El jugador 2 realizará las jugadas por medio de botones/switches de un Arduino.

5. Se debe realizar una pantalla inicial donde se establecerá cuál jugador inicia.
6. El Arduino se conectará a la FPGA por conexión serial (UART, SPI, I<sup>2</sup>C, etc).
7. Cada jugador tendrá un color asignado (definido por el grupo).
8. Si un jugador no hace su jugada en 10 segundos, el sistema le asignará una posición al azar, sobre las restantes. **Esta lógica se debe realizar con SystemVerilog.**
9. Por medio de 7 segmentos se mostrará el tiempo restante en el turno del jugador.
10. El juego deberá continuar hasta que un jugador logre una línea (vertical, horizontal o diagonal) de 4 de sus figuras, o hasta que no queden casillas libres.
11. Al finalizar el juego deberá mostrarse en pantalla el jugador ganador, así como resaltarse la jugada ganadora (línea).
12. Es creatividad de los estudiantes la representación de todos los elementos en el tablero. Sin embargo, debe ser intuitiva.
13. **Toda lógica de control se manejará por medio de la FSM, cualquier solución que no se realice de esta forma, tendrá una nota de 0.** Otros componentes combinacionales y secuenciales deben ser utilizados siguiendo un enfoque de diseño modular. La FSM deberá generar **únicamente** señales de control hacia los diferentes módulos del sistema.

### Detalles de implementación

Para la lógica del juego se debe diseñar una máquina de estados finitos (FSM) que permita llevar el **control** del movimiento, selección de casillas y las condiciones de victoria y derrota. La máquina deberá interactuar con diferentes módulos (contadores, temporizadores, decodificadores) con el fin de realizar las diferentes funciones de la aplicación. No se permite utilizar el *wait* (*#10*) de SystemVerilog como temporizador.

Finalmente, el sistema deberá contar con una lógica de reset, que vuelva al estado inicial al proceso, sin importar en qué sección se encuentre.

Con base en la descripción anterior:

1. Diseñe el controlador VGA. Parta de los circuitos básicos (contadores, comparadores, etc) que considere necesarios. Muestre los diagramas de bloques, tablas de verdad, diagramas de tiempos y circuitos de cada módulo en el diseño.

2. Diseñe la máquina de estados del controlador de la aplicación. Implemente el diseño utilizando SystemVerilog. Muestre el diagrama de estados del controlador.
3. Diseñe la lógica para detección de victoria, generación de las figuras, así como demás módulos necesarios. Utilice combinaciones de modelos de estructura y comportamiento para dichos módulos.
4. Integre el controlador VGA, la máquina de estados y demás módulos. Muestre un diagrama de bloques completo del diseño.
5. Realice un *testbench* de autochequeo, en SystemVerilog, en el que se muestre de manera simple el funcionamiento de la aplicación (controlador general + lógica necesaria).
6. Implemente en la FPGA el sistema completo.

## 4. Metodología

Se utilizará la metodología explicada en el laboratorio 1.

**Los documentos serán sometidos a control de plagios.** La entrega se debe realizar por medio del TEC-Digital en la pestaña de evaluación. No se aceptan entregas extemporáneas después de la fecha de entrega a las 11:59 pm como máximo.