

# Laboratorio 3: Lógica Secuencial y Controladores

1<sup>st</sup> Javier Tenorio Cervantes  
carnet: 2020065308

2<sup>nd</sup> Axel Flores Lara  
carnet: 2021453573

3<sup>th</sup> Kevin Lobo Juárez  
carnet : 2020087823

## I. ABSTRACT

Este informe se enfoca en el diseño de un controlador VGA y un juego de Tic-Tac-Toe utilizando lógica secuencial. Se estudian máquinas de estado finito (FSM), distinguiendo entre máquinas de Moore y Mealy, y se analizan conceptos como el tiempo de configuración y el rebote en señales digitales. Los estudiantes desarrollan un controlador para una resolución de 640x480 píxeles y diseñan un juego que permite la interacción entre dos jugadores. Finalmente, se implementa el sistema en una FPGA, aplicando conceptos de diseño digital.

**Index Terms**—Maquina de estados finitos, maquina de Moore, maquina de Healy, circuitos secuenciales, VGA

## II. INTRODUCCIÓN

Las maquinas de estado finitos están basadas en los circuitos secuenciales síncronos donde la salida del circuito depende de varios factores no únicamente de las entradas sino de los estados previos, los estados son almacenados en registros. La razón por la que se le conocen como síncronos es debido a uso de clock para el cambio de un estado [1].

Las maquinas finitas es la manera en la que se pueden representar estos circuitos, debido a que como componentes principales necesitan un clock, en algunos casos un reset para iniciar. Los registros vienen definidos por  $k$  eso quiere decir que es finito, por lo tanto, los estados están definidos por  $2^k$  [1].

Básicamente, el funcionamiento consiste de la siguiente manera, cuando hay un flanco de subida en el reloj, entonces la lógica de siguiente estado empieza a calcular el próximo estado, mientras que la lógica de salida calcula las salidas, por lo tanto, el próximo estado se almacena en los registros convirtiéndose así, en el estado actual de la maquina, este proceso se repite hasta encontrar una salida [1].

Existe la maquina de Moore y de Mealy, de hecho, la maquina de Mealy tiene un parecido a la maquina de Moore. Pero las diferencias son las siguientes:

**Salidas:** Las salidas en la maquina de Moore dependen únicamente del estado en el que maquina se encuentra. Mientras que una maquina de Mealy las salidas dependen del estado actual y de las entradas que recibe [1].

**Tiempos de salida:** Los cálculos del estado siguiente son toman un poco más de tiempo en el caso de la maquina de Moore debido a que calcula el próximo estado después de la

transición, mientras que en la maquina de Healy los cálculos son realizados en el momento de la transición [1].

**Número de estados:** La maquina de Healy puede representar lo mismo que una maquina de Moore una cantidad menor de estados [1].

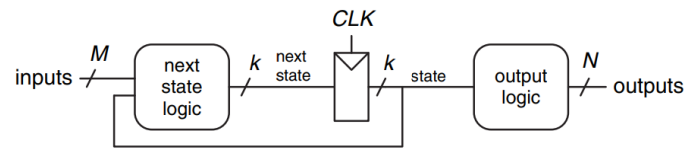


Fig. 1. Diagrama de señales maquina de Moore

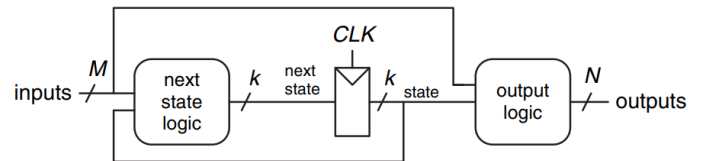


Fig. 2. Diagrama de señales maquina de Mealy

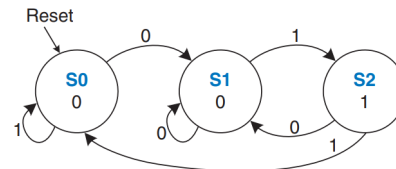


Fig. 3. Diagrama de estados maquina de Moore

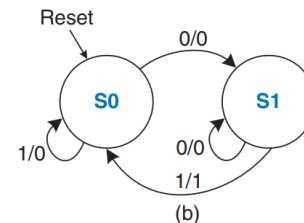


Fig. 4. Diagrama de estados maquina de Mealy

Seguidamente, dos conceptos fundamentales para comprender las máquinas de estado finito son el setup time y el hold

time. El setup time o tiempo de configuración se define como el período mínimo antes del borde activo del reloj durante el cual la señal de entrada debe permanecer estable para que los datos puedan ser capturados correctamente. Si los datos no están estables antes del borde activo del reloj, no se registrarán correctamente en el flip-flop, lo que puede causar errores en el almacenamiento de la información [1].

Por otro lado, el hold time o tiempo de retención se refiere al período mínimo después del borde activo del reloj durante el cual la señal de entrada debe permanecer estable. Es esencial que, después de este flanco, los datos continúen siendo estables para que el flip-flop capture y almacene correctamente la información [1].

El efecto de rebote es muy común en switches o botones mecánicos que se pueden considerar como interruptores, este problema ocurre cuando los interruptores se cierran o se abren no realizan un contacto único por lo que generan varios rebotes separándose y conectándose, generalmente los rebotes ocurren de 10 a 100 veces en un periodo de 1 milisegundo [2]. Esto puede provocar un comportamiento no esperado en el caso de contadores que podría tomar en cuenta a los pulsos como entradas validas.

El uso de un flip-flop SR es una de las soluciones para mitigar el efecto de rebote en interruptores mecánicos. Este circuito cuenta con dos entradas, Set (S) y Reset (R), las cuales son activas en bajo. Esto significa que el flip-flop se activa cuando las señales de estas entradas están en nivel bajo (0 lógico). Al accionar el interruptor, el flip-flop SR recibe la señal y la guarda en un estado estable ignorando oscilaciones del rebote, proporcionando una salida limpia y sin rebote. La configuración del circuito se puede visualizar en la Figura 5.

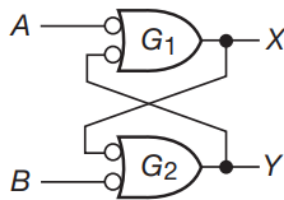


Fig. 5. Flip-flop SR

Por otro lado, la siguiente solución es utilizando inversores en cascadas, requiere únicamente de dos inversores en una configuración de cascada. Cuando se acciona el interruptor, lo que hacen los inversores es suavizar las transiciones, eliminando cualquier señal errática generada por el rebote esto debido a que interceptan la señal de manera gradual [2]. La configuración del circuito se puede visualizar en la Figura 6

En la VGA es posible encontrar una serie de señales que permiten el funcionamiento como lo es horizontal sync, vertical sync y RGB.

En el caso de **horizontal sync** o señal de sincronización horizontal consiste en la actualización de la señal de izquierda a derecha, esta señal es fundamental porque indica el final de una línea y el comienzo de la siguiente línea, para su

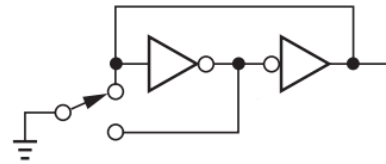


Fig. 6. Inversores en cascada

funcionamiento requiere de una tensión mayor o igual a 2.4V [3].

Por otro lado, en el caso de **vertical sync** o señal de sincronización vertical consiste en la frecuencia con la que el monitor actualiza la imagen completa, esta actualización va desde la parte superior de la pantalla a la parte inferior, indica el final de una imagen y el comienzo de la siguiente, para su funcionamiento requiere de una tensión mayor o igual a 2.4V [3].

Estas señales permiten una correcta actualización en las imágenes, ambas cuentan con barrido lo que permite limpiar la imagen anterior antes de colocar una nueva [3].

Además para brindar color a las imágenes VGA utiliza señales analógicas de **RGB (rojo, azul, verde)** estos son colores primarios, cuando se combinan generan una amplia variedad de colores, consiste en colorear individualmente cada píxel, para que las señales funcionen o se muestren activas requiere de 0.7V, menos de ese valor se considera como la señal está apagada [3].

A continuación se presenta el diagrama de señales para una configuración VGA 640x480 en la figura 7 se puede visualizar a lo largo del tiempo el comportamiento de la señal en horizontal, mientras que en la figura 8 el comportamiento de la señal en vertical.

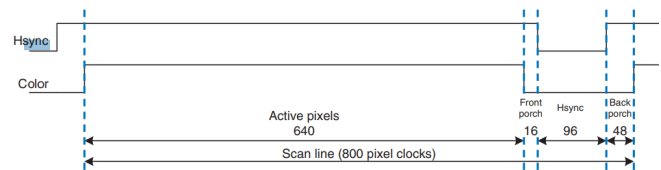


Fig. 7. Diagrama de tiempo de sincronización horizontal

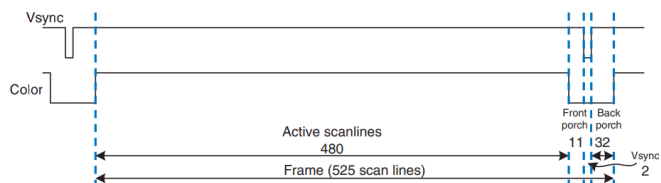


Fig. 8. Diagrama de tiempo de sincronización vertical

Para el cálculo de las frecuencias en las señales de sincronización horizontal y vertical, a continuación:

### Frecuencia de Sincronización Horizontal (Hsync)

La frecuencia de sincronización horizontal está relacionada con la duración de una línea completa de píxeles, que incluye tanto los píxeles visibles como los píxeles adicionales necesarios para la sincronización [3].

*Parámetros:*

- **Resolución horizontal:** 640 píxeles visibles por línea.
- **Total de píxeles por línea:** 800 píxeles (640 visibles + píxeles de porches y pulsos de sincronización) [3]:

$$\text{Total de píxeles por línea} = 640 + 16 + 96 + 48 = 800$$

- **Tiempo por píxel:** 39.72 ns.

*Cálculo de la duración de una línea completa:* La duración de una línea completa ( $T_{\text{línea}}$ ) es el producto del número total de píxeles y el tiempo por píxel [3]:

$$T_{\text{línea}} = 800 \text{ píxeles} \times 39.72 \text{ ns} = 31.776 \mu s$$

*Cálculo de la frecuencia horizontal*

La frecuencia de sincronización horizontal ( $f_H$ ) es el inverso de la duración de una línea completa [3]:

$$f_H = \frac{1}{T_{\text{línea}}} = \frac{1}{31.776 \times 10^{-6}} \approx 31.47 \text{ kHz}$$

Por lo tanto, la frecuencia de sincronización horizontal es aproximadamente 31.47 kHz.

### Frecuencia de Sincronización Vertical (Vsync)

La frecuencia de sincronización vertical está relacionada con la cantidad de cuadros que se refrescan por segundo en la pantalla [3].

*Parámetros:*

- **Resolución vertical:** 480 líneas visibles.
- **Total de líneas por cuadro:** 525 líneas (480 visibles + líneas de porches y pulsos de sincronización) [3]:

$$\text{Total de líneas por cuadro} = 480 + 11 + 2 + 32 = 525$$

- **Frecuencia de sincronización horizontal:** 31.47 kHz.

*Cálculo de la frecuencia vertical:* La frecuencia de sincronización vertical ( $f_V$ ) se obtiene dividiendo la frecuencia horizontal entre el número total de líneas por cuadro [3]:

$$f_V = \frac{f_H}{\text{Total de líneas}} = \frac{31.47 \text{ kHz}}{525} \approx 59.94 \text{ Hz}$$

**Frecuencia Hsync:** 31.47 kHz

**Frecuencia Vsync:** 59.94 Hz

Estas frecuencias aseguran que la pantalla se refresque correctamente en una resolución de 640x480 píxeles, a una tasa de refresco de 60 cuadros por segundo aproximadamente [3].

A continuación una propuesta del controlador VGA en la figura 9.

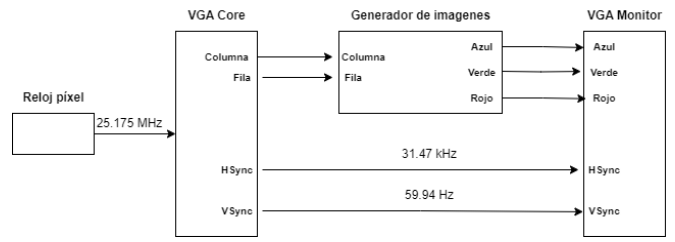


Fig. 9. Propuesta diagrama de bloques controlador VGA

## III. DESARROLLO

*Problema 3.1*

*Problema 3.2*

*Problema 3.3*

## IV. RESULTADOS

## V. CONCLUSIONES

## REFERENCES

- [1] Sarah Harris y David Harris. Digital design and computer architecture: arm edition. Morgan Kaufmann, 2015
- [2] P. Horowitz y W. Hill, The Art of Electronics, 3.a ed. 2015.
- [3] K. Jack, Video demystified, 4.a ed. 2005.