

# Laboratorio #4: Lógica Secuencial de Control

Jorge Agüero Zamora, Arturo Chinchilla Sánchez, Luis Murillo Rios  
 georgeaz56@gmail.com mchinchilla11@gmail.com luismuelino@gmail.com  
 Área Académica de Ingeniería en Computadores  
 Instituto Tecnológico de Costa Rica

**Resumen**—El siguiente documento trata sobre el diseño y creación de una máquina de café utilizando sistemas secuenciales, mediante la utilización de una máquina de estados finitos que maneja los diferentes estados de la preparación del café. También se mencionan temas importantes a la hora de diseñar un sistema secuencial, como el rebote de señales digitales en sistemas mecánicos, el *setup* y *hold* time y el uso de un divisor de frecuencia para manejar tiempos de la máquina de estados.

**Palabras clave**—Circuito secuencial, máquina de estados finitos, *setup* y *hold* time, divisor de frecuencia

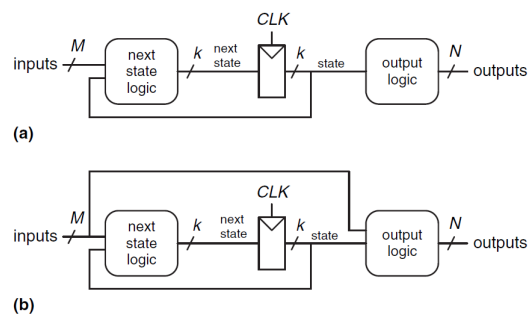


Figura 1. Máquinas de estado finito: (a) Máquina de Moore, (b) Máquina de Mealy (Imagen tomada de [1])

## I. INTRODUCCIÓN

Según [1] un circuito secuencial es aquel en el que sus salidas dependen de los valores actuales y anteriores de las entradas; en otras palabras, depende de la secuencia de entrada. Una de las diferencias con respecto a los circuitos combinatoriales es que los secuenciales sí tienen memoria, ya que como se menciona anteriormente dependen de los valores anteriores de las entradas (*memoria del pasado*).

Para [1] una Máquina de Estados Finitos (*FSM* por sus siglas en inglés) son circuitos secuenciales síncronos que tiene  $M$  entradas,  $N$  salidas y  $k$  bits de estado. También recibe un reloj y, opcionalmente, una señal de reinicio (*reset*).

Una *FSM* consta de dos bloques de lógica combinatorial, lógica de salida y estado siguiente dado el estado actual y la entrada, y un registro que almacena el estado. Para cada ciclo del reloj, la máquina pasa al estado siguiente calculado usando el estado actual y las entradas.

Existen dos tipos generales de *FSM*, que se caracterizan por sus especificaciones funcionales, las llamadas Máquinas de Moore, donde las salidas dependen solamente de el estado actual de la máquina, y las Máquinas de Mealy, donde las salidas dependen tanto del estado actual de la máquina como de las entradas de corriente. En la figura 1 se muestran las formas para ambos tipos de máquinas.

En el diseño de sistemas digitales lógicos se debe tener en cuenta aspectos como el rebote de señales

digitales por elementos mecánicos, esto sucede por ejemplo con dispositivos como botones, switches, etc. Este efecto de rebote consiste en que al cerrar una conexión mediante estos dispositivos las láminas utilizadas como contactos chocan, generando una fuerza normal en sentido contrario, separando las placas, y así una y otra vez hasta que esta energía se disipe, ocasionando que la señal empiece a brincar entre un estado lógico de 1 y 0, antes de terminar estabilizándose. Sin embargo, puede ser perjudicial para nuestros sistemas que esto suceda, ya que detectará todos los cambios en las entradas y puede generar un comportamiento no deseado.

Dos maneras comúnmente usadas para resolver este problemas son, por Hardware y Software, para la primera se hace con el uso de un circuito RC, el cual, aunque no evita el rebote, logra que la oscilación no vaya entre "cerrado" y "abierto". Mediante software se pueden utilizar técnicas como la llamada "cambio de estado demorado" que consiste en cambiar el estado de la entrada conectada al dispositivo generador del rebote un tiempo "X" después que se haya detectado el primer cambio e ignorar todas las oscilaciones intermedias ocurridas. Durante este laboratorio se utilizó un circuito 'sincronizador' para mitigar el efecto de rebote.

Se define como "*setup time*" a la la cantidad de tiempo mínima antes del límite activo del reloj para que los datos sean estables y se bloqueen A, y

"hold time" como la cantidad mínima de tiempo después del borde activo del reloj durante el cual los datos deben ser estables.

Este documento está estructurado de la siguiente manera, sección de resultados, sección de análisis de los resultados obtenidos, conclusiones y las referencias utilizadas durante este laboratorio.

## II. RESULTADOS

Para el diseño de la máquina de café se hicieron dos máquinas de estado, una encargada de realizar las verificaciones de dinero y la selección de bebida. Y otra máquina de estados encargada de la preparación del café y los tiempo por ingrediente del mismo.

La primera maquina para la selección de bebida cuenta con tres estados los cuales son:

- **Estado 0:** Estado en el cual no hay monedas, la máquina se quedará en este estado hasta que comience a entrar dinero.
- **Estado 1:** En este estado ya se presenta dinero en la maquina, pero todavía no se ha seleccionado una bebida. En el momento en que se seleccione una bebida, se entra a un módulo el cual verifica que haya suficiente dinero para la bebida seleccionada. Si este es el caso se pasa al Estado 2. Se debe tomar en cuenta que la máquina de café presenta un botón de cancelar como especificación de diseño el cual cancela la bebida y devuelve el dinero depositado, en este estado es la última vez que se se podrá cancelar la bebida al pasar al siguiente estado esta opción no estará disponible.
- **Estado 2:** Este estado es la unión entre las dos máquinas de estado, después de verificar que la cantidad de dinero sea suficiente para la bebida seleccionada se pasa a la máquina encargada de realizar la bebida, mientras se crea la bebida esta máquina de estados se detendrá en el estado actual hasta que le llegue la bandera de finalizar bebida, la cual se da cuando la bebida se encuentra preparada.

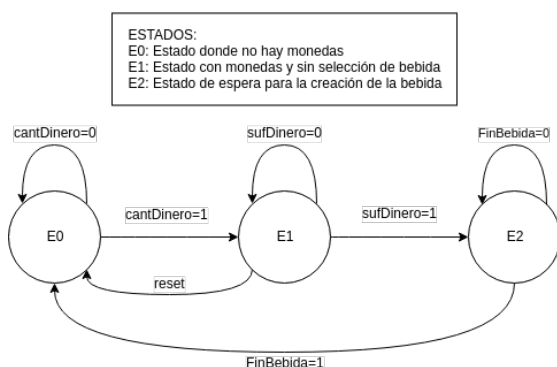


Figura 2. Máquina de estados finitos para la selección de bebida.

Para lograr su cometido, la máquina de control se apoya de un módulo que lleva la suma de las monedas que se han ingresado, el cuál es un simple contador que suma la cantidad correspondiente a las dos posibles monedas según sea la entrada, así como otro módulo que le indica si la bebida seleccionada por el usuario puede ser comprada con el dinero ingresado.

Este último módulo también se encarga de calcular el vuelto, puesto que para saber si se tiene suficiente dinero, se resta la cantidad de monedas del contador con el precio que se tiene conectado directamente en un MUX, codificado en 4 bits.

Para el caso de las entradas de monedas, que fueron modelados con switches, así como para la entrada de selección de bebida, se utilizó un circuito "sincronizador", que consta de dos registros en serie, para poder mitigar el efecto de rebote en las señales, así como minimizar la posibilidad de estados de "meta – estabilidad" para la máquina por desfases en el hold time o set up time causados por estas entradas.

La segunda máquina de estados es la encargada de la preparación de la bebida. Esta cuenta con estados intermedios entre cada estado de preparación de ingredientes, estos son los estados R. Los mismos se encuentran para dar el "reset" para el módulo encargado del cálculo de tiempo, se utilizan ya que se trabaja con un circuito secuencial y el clock de preparación es distinto al clock del módulo de tiempo por ingrediente, estos estados intermedios permiten que el clock se mantenga sincronizado.

Los estados para la maquina de preparación son los siguientes.

- **Estado B0:** Estado inicial de la máquina de preparación de bebida, este estado espera la señal suficiente dinero de la máquina de selección de bebida para pasar al estado "A" y comenzar la preparación.
- **Estado A:** Estado encargado de agregar el agua para la preparación de la bebida.
- **Estado C:** Estado encargado de agregar el café para la preparación de la bebida.
- **Estado L:** Estado encargado de agregar el azúcar para la preparación de la bebida.
- **Estado Ch:** Estado encargado de agregar el azúcar para la preparación de la bebida.
- **Estado Az:** Estado encargado de agregar el azúcar para la preparación de la bebida.
- **Estados Rx:** Funcionan como estados intermedios para que el módulo que calcula el tiempo de cada ingrediente ponga todas sus señales en cero y así pueda calcular el tiempo para el siguiente ingrediente, esto evita que se lean señales incorrectas desde este módulo.

Como se lee en la especificación del laboratorio, para cada tipo de bebida se difiere entre los ingredientes utilizados y la cantidad de los mismos que

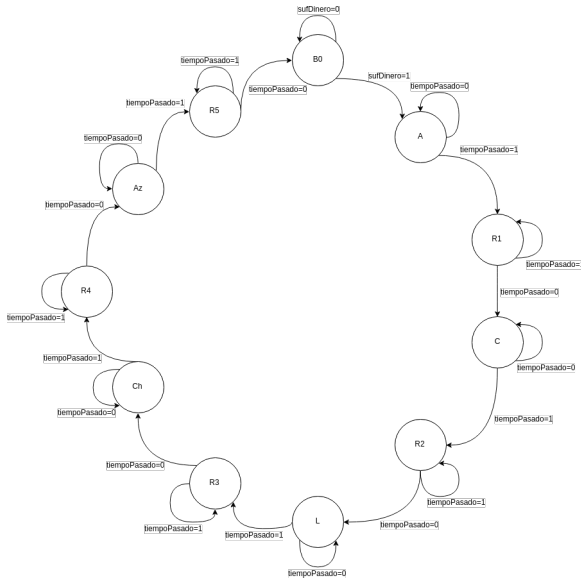


Figura 3. Máquina de estados finitos para preparación de bebida.

debe ser agregada (el tiempo en que cada electroválvula está activa).

La máquina de preparación se apoya del módulo que calcula el tiempo necesario según el ingrediente que se está sirviendo y según la bebida seleccionada. Este circuito funciona dividiendo la frecuencia del clock interno de la FPGA en la que se implementó (50MHz) mediante el uso de un contador, en el cual se calculó el bit que divide la frecuencia para obtener las necesarias de 1s (1 Hz), 2s (0,5 Hz) y 3s (0,33 Hz) con la fórmula

$$n = \log_2\left(\frac{F_{FPGA}}{F_0}\right) - 1$$

Donde  $F_0$  es la frecuencia deseada.

### III. ANÁLISIS DE RESULTADOS

Al ser los precios de las bebidas todos múltiplos de 100, igual que las posibles monedas, solamente se debió codificar el dígito más significativo de las cifras de suma de monedas y de cálculo de vuelto, por lo que estos fueron codificados utilizando 4 bits, permitiendo una cantidad máxima de dinero de 1400 que se puede contar, aunque el máximo monto que se podrá mostrar será 900, puesto que se limitó a usar 3 pantallas de 7 segmentos, uno mostrando la cifra codificada, y otros dos mostrando 0 para formar las centenas. Esto facilitó y redujo la cantidad de bits necesarios para la construcción de la máquina, lo cual implica menos hardware.

Las bebidas fueron codificadas en 'one hot', de la manera mostrada en el Cuadro I, lo cual simplifica la lógica necesaria para el cálculo de tiempo por ingrediente durante la preparación de la bebida y

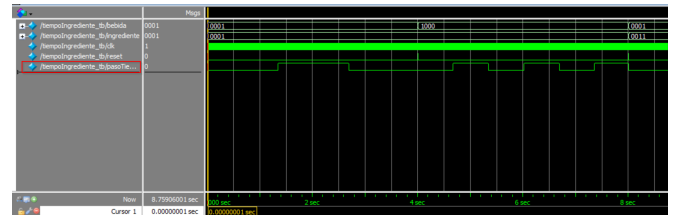
Cuadro I  
CODIFICACIÓN DE SELECCIÓN DE BEBIDA CON 4 BITS

| Bebida         | Codificación |
|----------------|--------------|
| Café expreso   | 0001         |
| Café con leche | 0010         |
| Capuccino      | 0100         |
| Mocaccino      | 1000         |
| Sin selección  | 0000         |

al no haber demasiados tipos de bebidas el tamaño de la codificación con 4 bits es razonable.

El funcionamiento de la máquina fue probado mediante la implementación de archivos de *testbench* en los que se simulan las posibles entradas a la máquina y se analizan sus salidas, probándolos con las salidas esperadas.

Durante la creación de los *testbench* se debió modelar un *clock* de 50 MHz para probar el correcto funcionamiento del divisor de frecuencia, el cual provee los tiempos para cada ingrediente. La figura III muestra una simulación de un *testbench* de este circuito, donde la señal *pasoTiempo*, cambia su frecuencia según la selección de bebida y el ingrediente utilizado.

Figura 4. Prueba de cambio de frecuencia en *pasoTiempo* según selección de bebida e ingrediente

### IV. CONCLUSIONES

Las FSM son una herramienta útil para el modelado de problemas secuenciales, y los lenguajes de HDL como SystemVerilog proveen una forma sencilla de construir soluciones a estos problemas.

Los contadores proveen un mecanismo sencillo para poder dividir la frecuencia de un *clock* externo para así obtener las frecuencias deseadas.

Los circuitos que reciben entradas desde dispositivos físicos deben tener en cuenta los conceptos del efecto de rebote, así como el *set up time* y el *hold time*, e implementar medidas para mitigar estos efectos y cumplir los requerimientos de tiempo, para poder garantizar el funcionamiento correcto de estos circuitos. Los circuitos 'sincronizadores' proveen una forma sencilla de solucionar estos problemas.

### REFERENCIAS

- [1] S. Harris and D. Harris, *Digital design and computer architecture*. Amsterdam [i 11 pozostałych]: Elsevier / Morgan Kaufmann Publishers, 2016.