

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0523 - Circuitos Digitales II

I ciclo 2023

Tarea 3

## Diseño de un controlador para cajero automático

Isaí David Vargas Ovarés B88263

Grupo 1

Profesor: Enrique Coen Alfaro

7 de mayo de 2023

# 1. Resumen

Se implementa un cajero por medio de una maquina de estados, donde se inicia con el usuario insertando su tarjeta bancaria en el cajero automático. El cajero automático solicita al usuario que introduzca su número de identificación personal (PIN) para verificar su identidad. El usuario ingresa su PIN y el cajero lo confirma. Si el código PIN es incorrecto, el cajero automático no otorgará acceso a la cuenta. Una vez verificada la identidad del usuario, el cajero automático tiene dos opciones: retiro de efectivo y depósito. Si el usuario desea retirar efectivo, el cajero retira la cantidad solicitada, asegurándose de que haya saldo suficiente en la cuenta. Si el usuario desea realizar un depósito, el cajero automático recibe dinero y lo deposita en la cuenta del usuario. Esto basado en lo mencionado en [1], los circuitos con bloques lógicos combinacionales y secuenciales pueden describirse como máquinas de estados finitos o simplemente como máquinas de estados finitos. Las máquinas se sincronizan cuando los cambios de estado activan las señales del reloj. Una máquina cuyo funcionamiento no depende de una señal de reloj se denomina asíncrona. El estado actual (EA) del dispositivo está determinado por las variables ocultas de los flip-flops de la parte secuencial (o biestabilidad). El siguiente estado de la máquina (EP) está determinado por el circuito de la parte de composición. En máquinas de estados finitos, podemos distinguir entre modelos de Moore y modelos de Mealy

# 2. Descripción Arquitectónica

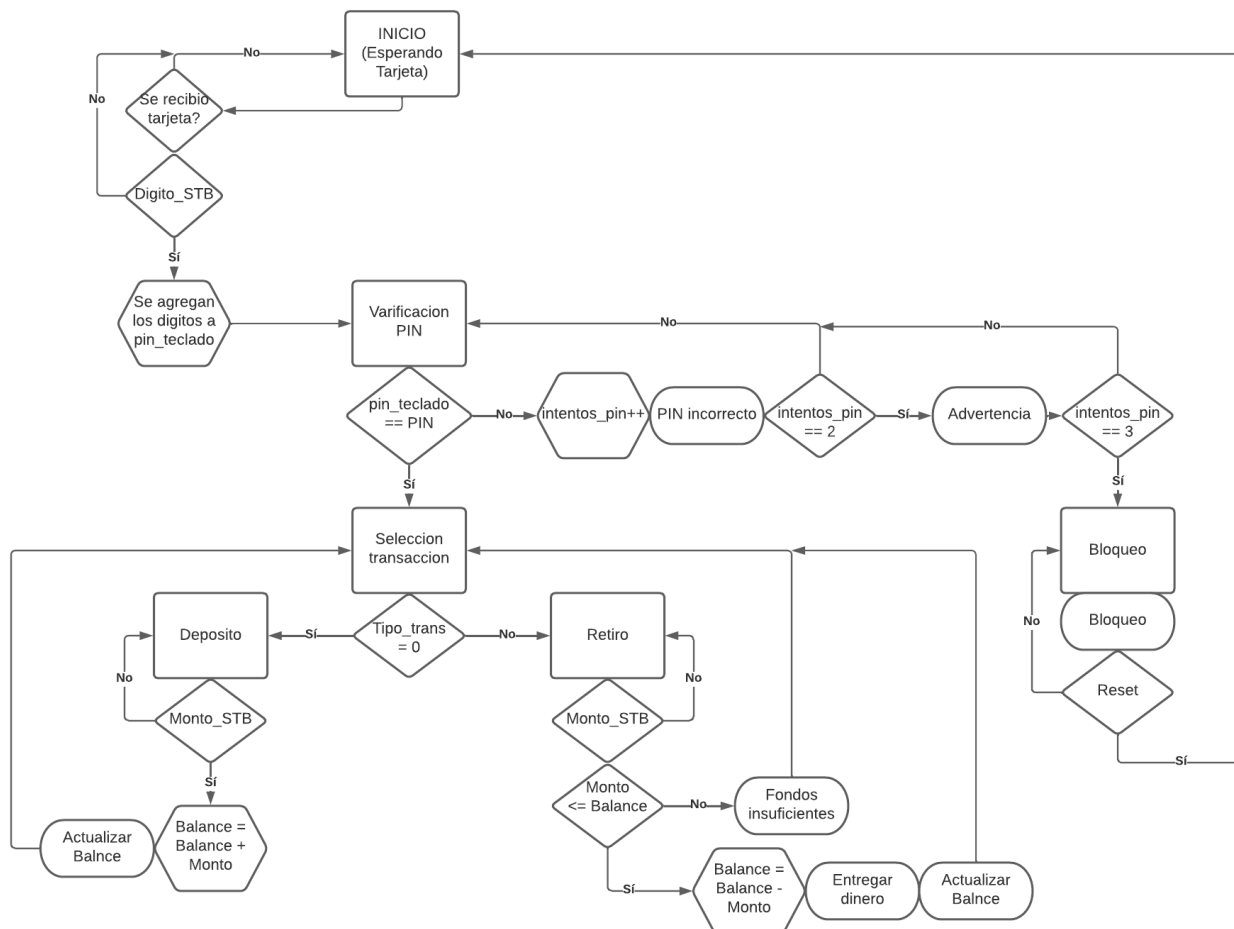


Figura 1: Diagrama ASM de la maquina de estados.

Los cajeros automáticos tienen múltiples entradas y salidas y dependen de máquinas de estado para controlar su funcionamiento. A continuación se detalla la documentación de entrada y salida del módulo, así como el funcionamiento de la máquina de estados.

```

1  module cajero (CLK,Reset,Tarjeta_recibida,PIN,Digito,Digito_STB,Tipo_trans,Monto,Monto_STB,
2      Balance_actualizado,Entregar_dinero,Fondos_insuficientes,PIN_incorrecto,Advertencia,Bloqueo
3      );
4      input CLK,Reset,Tarjeta_recibida,Digito_STB,Tipo_trans,Monto_STB;
5      input [3:0] Digito;
6      input [15:0] PIN;
7      input [31:0] Monto;
8
9      output reg Balance_actualizado,Entregar_dinero,Fondos_insuficientes,PIN_incorrecto,Advertencia,Bloqueo;
10
11
12      // Estados
13      parameter ESPERA_TARJETA = 3'b000; //valor que representa el estado inicial de la máquina de estados (000).
14      parameter VERIFICACION_PIN = 3'b001; //valor que representa el estado en el que se verifica el número PIN ingresado (001).
15      parameter SELECCION_TRANSACCION = 3'b010; //valor que representa el estado en el que se selecciona el tipo de transacción a realizar (010).
16      parameter DEPOSITO = 3'b011; //valor que representa el estado en el que se realiza una transacción de depósito (011).
17      parameter RETIRO = 3'b100; //valor que representa el estado en el que se realiza una transacción de retiro (100).
18      parameter BLOQUEADO = 3'b101; //valor que representa el estado en el que la tarjeta ha sido bloqueada (101).

```

Figura 2: Entradas, salidas y estados parametrizados la maquina de estados.

## 2.1. Entradas

- CLK: señal de reloj que controla el ciclo de operación del módulo.
- Reset: señal de reset que se utiliza para poner el módulo en un estado conocido y estable al principio de su funcionamiento.
- Tarjeta\_recibida: señal que indica si se ha recibido una tarjeta válida.
- Digito\_STB: señal de strobe para el ingreso de un dígito en el teclado.
- Tipo\_trans: señal que indica el tipo de transacción que se desea realizar (depósito o retiro).
- Monto\_STB: señal de strobe para el ingreso del monto de la transacción.
- Digito: valor numérico del dígito ingresado en el teclado.
- PIN: valor numérico del PIN de la tarjeta.
- Monto: valor numérico del monto de la transacción.

## 2.2. Salidas

- Balance\_actualizado: señal que indica si el balance de la cuenta ha sido actualizado después de una transacción.
- Entregar\_dinero: señal que indica si se debe entregar el dinero solicitado en una transacción de retiro.
- Fondos\_insuficientes: señal que indica si no hay suficiente dinero en la cuenta para realizar la transacción de retiro solicitada.
- PIN\_incorrecto: señal que indica si el PIN ingresado en el teclado no coincide con el PIN de la tarjeta.
- Advertencia: señal que indica si se ha excedido el número máximo de intentos de ingresar el PIN incorrectamente.

- Bloqueo: señal que indica si la tarjeta ha sido bloqueada debido a que se ha excedido el número máximo de intentos de ingresar el PIN incorrectamente.

## 2.3. Estados

- ESPERA\_TARJETA: estado inicial en el que el módulo espera a que se inserte una tarjeta.
- VERIFICACION\_PIN: estado en el que se verifica el PIN de la tarjeta ingresado en el teclado.
- SELECCION\_TRANSACCION: estado en el que se selecciona el tipo de transacción a realizar.
- DEPOSITO: estado en el que se realiza una transacción de depósito.
- RETIRO: estado en el que se realiza una transacción de retiro.
- BLOQUEADO: estado en el que la tarjeta ha sido bloqueada.

```

21 // Declaración de los registros internos
22 reg [63:0] Balance; //registro de 64 bits que representa el balance actual del usuario.
23 integer intentos_pin; //registro de enteros que lleva la cuenta de los intentos de PIN incorrectos.
24
25
26 // Declaración de las señales internas
27 reg [15:0] PIN_teclado; //registro de 16 bits que almacena el número PIN ingresado por el usuario en el teclado.
28
29
30 // Definición de la máquina de estados
31 reg [2:0] estado_actual;
32 reg [2:0] estado_prox;

```

Figura 3: Señales internas y registros del cajero.

## 2.4. Señales internas

- PIN\_teclado: un registro de 16 bits que almacena el número PIN ingresado por el usuario en el teclado.
- estado\_actual y estado\_prox: registros de 3 bits que representan el estado actual y el próximo estado en la máquina de estados finitos.
- intentos\_pin: un registro entero que lleva la cuenta de los intentos de PIN incorrectos.
- Balance: un registro de 64 bits que representa el balance actual del usuario.

## 2.5. Definición de los flip flops

- Este es el estado inicial de la máquina de estado, que se activa en cada flanco ascendente del reloj (POSGE CLK).
- Se verifica si la señal de Reset está activa. Si es así, se configura el estado\_actual en ESPERA\_TARJETA e inicialice todas las demás señales en cero.

- Si la señal de Reset no está activa, el estado actual se establece en estado\_prox (un estado definido en otra parte del código), y se realiza una operación de registro de desplazamiento para almacenar el PIN ingresado en el teclado. El registro de desplazamiento se activa solo cuando se detecta una señal de "STB" (Strobe) de entrada en la línea de "Digito\_STB".

```

34     always @ (posedge CLK) begin
35         //Con en Reset se envían todos los valores a 0
36         if (Reset) begin
37             estado_actual      <= ESPERA_TARJETA;
38             Balance_actualizado <= 1'b0;
39             Entregar_dinero     <= 1'b0;
40             Fondos_insuficientes <= 1'b0;
41             PIN_incorrecto      <= 1'b0;
42             Advertencia         <= 1'b0;
43             Bloqueo             <= 1'b0;
44             PIN_teclado         <= 16'b0;
45             Balance             <= 64'b0;
46             intentos_pin        = 0;
47         end
48         else begin
49             estado_actual <= estado_prox;
50             //Registro desplazante para guardar los dígitos del PIN
51             if (Digito_STB) begin
52                 PIN_teclado[3:0] <= Digito;
53                 PIN_teclado[7:4] <= PIN_teclado[3:0];
54                 PIN_teclado[11:8] <= PIN_teclado[7:4];
55                 PIN_teclado[15:12] <= PIN_teclado[11:8];
56             end
57         end
58     end

```

Figura 4: Definición de los flip flops.

## 2.6. Sección combinacional

En la sección combinacional, se utiliza una estructura case para definir la lógica combinacional en función del estado actual de la máquina de estados y de las entradas del sistema. La lógica combinacional se encarga de determinar el próximo estado de la máquina de estados y las señales de salida en función del estado actual y de las entradas del sistema.

- En el estado ESPERA\_TARJETA, el diseño espera que se reciba una tarjeta. Si se recibe una tarjeta, se cambia al estado VERIFICACION\_PIN, si no se recibe una tarjeta, se mantiene en el estado ESPERA\_TARJETA.
- En el estado VERIFICACION\_PIN, el diseño verifica el código PIN ingresado por el usuario. Si el código PIN ingresado es incorrecto, se activa la señal PIN\_incorrecto y se aumenta el número de intentos realizados en la variable intentos\_pin. Si se ingresó incorrectamente el código PIN 8 veces, en este caso 8 por el pin cambia 4 veces, o sea una vez correspondiente a cada introducción de Dígito, se activa la señal Advertencia y se mantiene en el estado VERIFICACION\_PIN. Si se ingresó incorrectamente el código PIN 12 veces, se activa la señal Bloqueo y se cambia al estado BLOQUEADO. Si el código PIN ingresado es correcto, se desactiva la señal PIN\_incorrecto y se cambia al estado SELECCION\_TRANSACCION.
- En el estado BLOQUEADO, el diseño espera a que se reciba la señal de reset para desbloquearse. Si la señal de reset está activa, se cambia al estado ESPERA\_TARJETA. Si la señal de reset no está activa, se mantiene en el estado BLOQUEADO.
- En el estado SELECCION\_TRANSACCION, el diseño espera a que el usuario seleccione una transacción. Si la variable Tipo\_trans es igual a 0, se cambia al estado DEPOSITO. Si

la variable Tipo\_trans es igual a 1, se cambia al estado RETIRO. Si la variable Tipo\_trans es cualquier otro valor, se mantiene en el estado SELECCION\_TRANSACCION.

- En el estado DEPOSITO, el diseño espera a que el usuario ingrese una cantidad de dinero y presione el botón "Listo.<sup>en</sup> el teclado. Si la señal Monto\_STB es activa, se actualiza el saldo de la cuenta bancaria sumando el monto ingresado a la variable Balance y se activa la señal Balance\_actualizado. Si la señal Monto\_STB no está activa, se desactiva la señal Balance\_actualizado y se mantiene en el estado DEPOSITO.
- En el estado RETIRO, el diseño espera a que el usuario ingrese una cantidad de dinero y presione el botón "Listo.<sup>en</sup> el teclado. Si la señal Monto\_STB es activa y la cantidad ingresada es menor o igual al saldo disponible en la variable Balance, se actualiza el saldo de la cuenta bancaria restando el monto ingresado a la variable Balance, se activa la señal Balance\_actualizado y se activa la señal Entregar\_dinero.

### 3. Plan de Pruebas

Se realizaron las mismas pruebas que en la tarea 1. Adicionalmente se realizaron las siguientes pruebas al proceso de síntesis.

- Prueba 1, para esta primera prueba se introduce el PIN incorrecto 3 veces, esto para corroborar el funcionamiento de las señales de advertencia, bloqueo y el estado de bloqueado.
- Prueba 2, en este caso se introduce el PIN correcto y se elige realizar un depósito, para corroborar el comportamiento de la señal de balance actualizado y si este se actualizó, posteriormente se elige realizar un retiro de una cantidad menor a la depositada, o sea que se retira una cantidad menor al balance, en este caso se analizan las señales de balance actualizado y entregar dinero, luego se vuelve a seleccionar realizar un retiro pero ahora de una cantidad mayor al balance para corroborar la señal de fondos insuficientes.

### 4. Instrucciones de utilización de la simulación

Esta tarea fue desarrollada en el entorno de EDAPlayground por lo que basta con ingresar al siguiente link y correr la simulación.

- <https://www.edaplayground.com/x/Vezb>

### 5. Resultados

- **Prueba 1:** Se introduce el PIN incorrecto por primera vez, esto se puede ver en los pulsos de Digito\_STB, a su vez se va moviendo el valor de PIN\_teclado, que como se puede ver en ningún momento coincide con PIN, por lo que nunca se baja la señal de pin incorrecto, debido a esto no se pasa al siguiente estado, cuando se introduce por segunda vez el PIN incorrecto se tiene un resultado similar, sin embargo al terminar de ingresar este se levanta Advertencia, se introduce por tercera vez el PIN incorrecto, lo cual al terminar de introducir este provoca que se levante la señal de bloqueo y se pasa al estado de Bloqueado, por lo que se envía un Reset para volver el estado de esperando tarjeta, pero como la tarjeta sigue activa se pasa inmediatamente al estado de verificación de PIN.

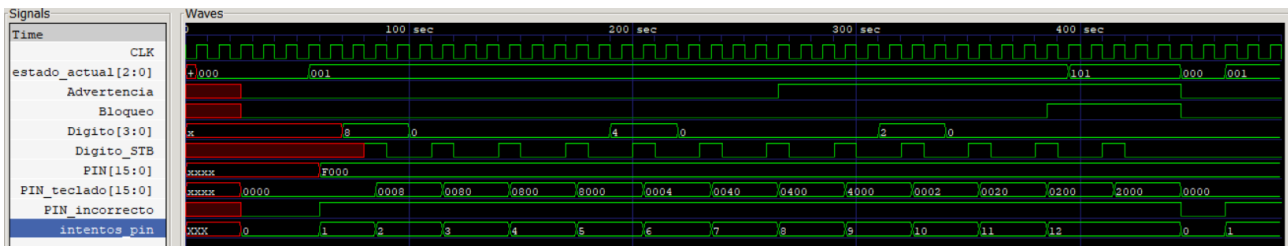


Figura 5: Prueba 1. Diagramas de tiempo.

- **Prueba 2:** En este caso se ingresa el PIN correcto, se puede ver que la señal de pin incorrecto se mantiene hasta que se ingresa en su totalidad el pin, esto es debido al tiempo que le toma al registro desplazante completarse, se ve como Dígito\_STB se levanta 4 veces correspondientes a cada dígito, en el estado actual se puede ver que se inicia en esperando tarjeta (000), se pasa a Verificación PIN (001), luego selección de transacción (010) donde primero se selecciona Deposito, por lo que se puede confirmar que se actualiza el balance y la señal de este, luego se selecciona Retiro y se retira una cantidad menor al balance por lo que actualiza este y se levanta la señal de entregar dinero, posteriormente se pide hacer un retiro mayor al balance, por lo que es te no se puede realizar y se levanta únicamente la señal de fondos insuficientes.

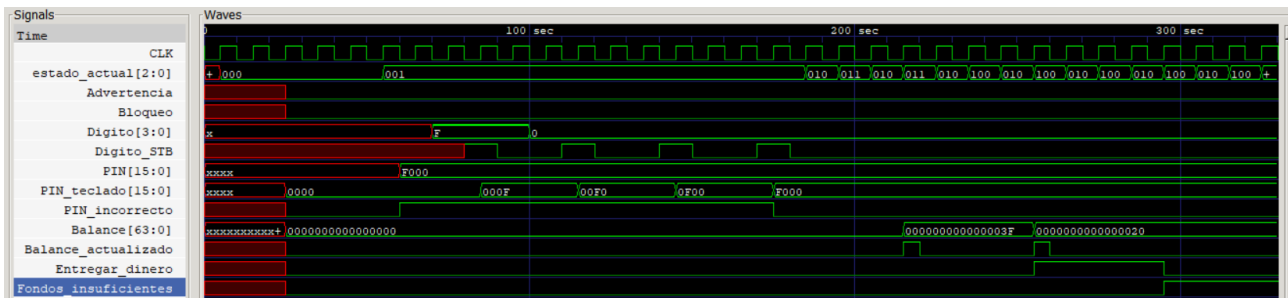


Figura 6: Prueba 2. Diagramas de tiempo.

## 6. Conclusiones

- En este caso se tuvo gran complicación con el registro desplazante para guardar los dígitos del PIN, ya que este se actualiza con el reloj y genera que se compare el PIN digitado con el PIN almacenado varias veces, antes de llegar al caso de match, este es un aspecto que se podría mejorar o corregir el diseño, ya que este podría derivar en falsos positivos.
- Se tiene un correcto funcionamiento de las señales de salida y estos podrían ser de utilidad, ya que en un caso real estos podrían ir conectados a luces o zumbadores, para indicar el estado actual, el balance, si el pin fue correteo o no.
- Con el uso de la sentencia case, se puede simplificar mucho el comportamiento de la maquina de estados, ya que otra manera se tendrían que utilizar muchos if.
- Utilizar los estados parametrizados con palabras relacionadas a la acción de cada uno ayuda mucho, ya que esto permite ver claramente el estado en que se encuentra el siguiente.

- Los cambios de los estados se realizan según un reloj, por lo que se trata de una maquina sincrónica, como se menciona en [\[1\]](#)

## Referencias

- [1] T. Ndjountche, *Electrónica digital 3*. ISTE Internacional, 2020.