

# Tarea 3

Mario Enrique Brenes Arroyo  
Universidad de Costa Rica  
Fecha: 7 de octubre de 2024  
Carnet: C11194  
Grupo: 1  
Curso: Circuitos Digitales II  
Profesor(a): Enrique Coen Alfaro

## Índice

<b>1. Resumen</b>	<b>2</b>
<b>2. Descripción Arquitectónica</b>	<b>3</b>
<b>3. Plan de Pruebas</b>	<b>6</b>
<b>4. Resultados</b>	<b>7</b>
<b>5. Casos de Prueba</b>	<b>7</b>
5.1. Caso 1: Depósito Perfecto . . . . .	7
5.2. Caso 2: Retiro Perfecto . . . . .	7
5.3. Caso 3: PIN Incorrecto Tres Veces . . . . .	8
5.4. Caso 4: Retiro Fallido (Fondos Insuficientes) . . . . .	8
<b>6. Resumen de Comandos</b>	<b>8</b>
<b>7. Conclusiones y recomendaciones</b>	<b>9</b>

# 1. Resumen

Para esta tarea se va a crear un modulo en verilog que consista en un cajero automático en donde este ahora tendrá que ser sintetizado para que se vea tanto estructural como conductualmente, ya que se verificarán tecnicas como el uso de librerias para simular componentes que el circuito tendria en la vida real.

Para este proyecto se implementará un digito un monto STB en donde funcionará como un botón para evaluar entradas del modulo en verilog.

Todo el proyecto lo puede ver en el repositorio de gihub [aquí](#)

## 2. Descripción Arquitectónica

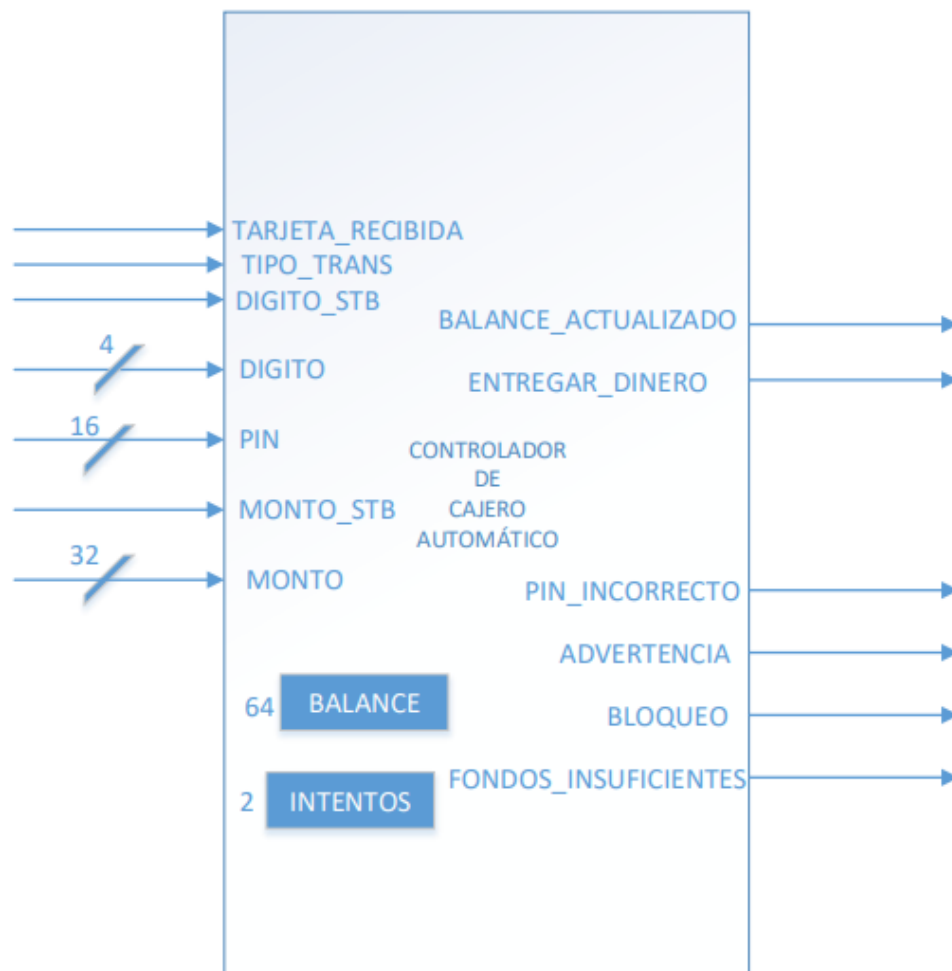


Figura 1: controlador para cajero automático

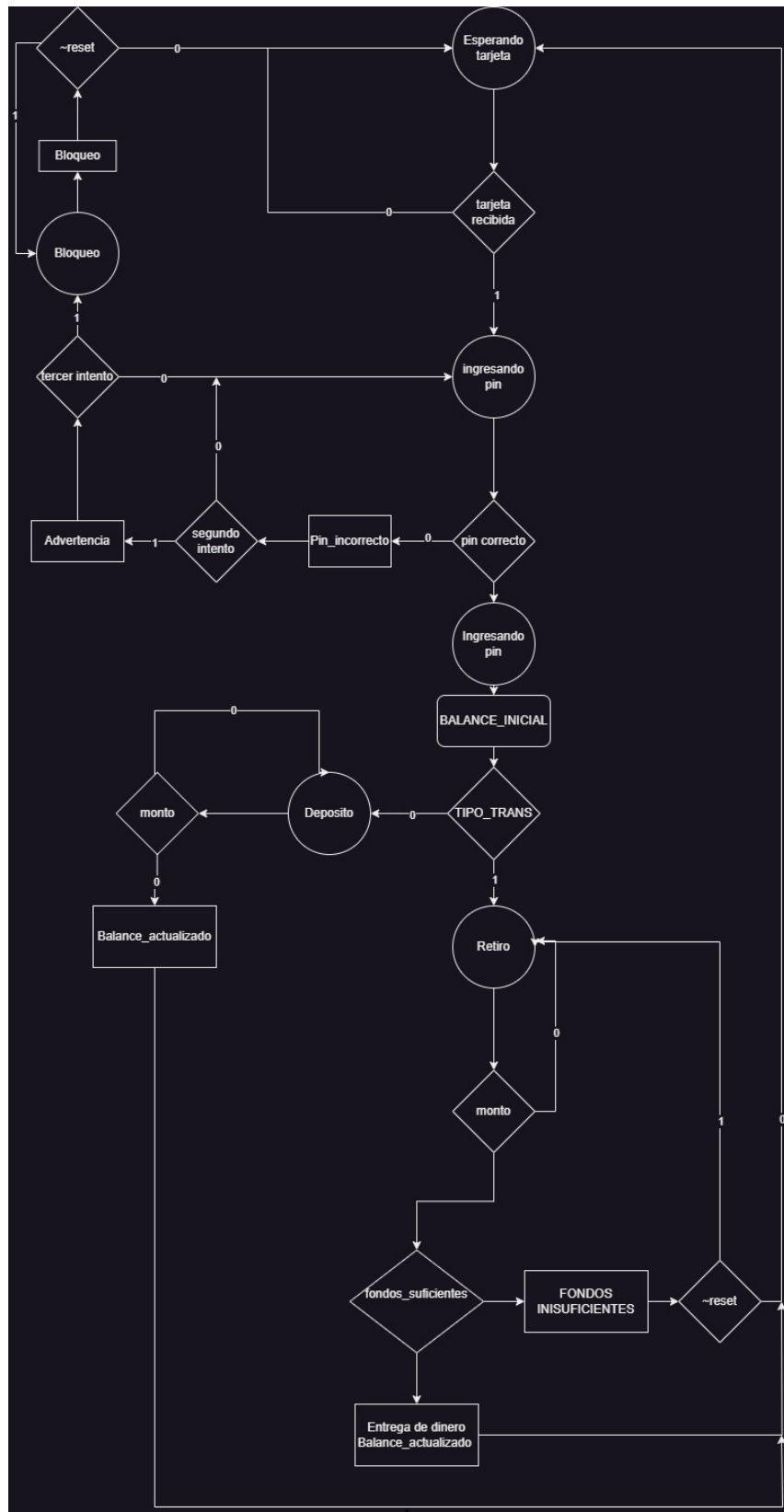


Figura 2: diagrama cajero automatico

■ **Entradas principales:**

- **TARJETA\_RECIBIDA:** Indica que se ha insertado una tarjeta.
- **PIN:** El PIN asociado con la tarjeta.
- **DIGITO y DIGITO\_STB:** El último dígito del PIN ingresado y una señal de strobe que indica que el dígito ha sido ingresado.
- **TIPO\_TRNANS:** Indica si la transacción es un depósito (0) o un retiro (1).
- **MONTO:** El monto de la transacción.
- **MONTO\_STB:** Indica que el monto ha sido ingresado.

■ **Estados de la FSM:**

- **Esperando\_tarjeta:** El sistema espera la inserción de una tarjeta (**TARJETA\_RECIBIDA**). Una vez detectada, pasa al estado de **Ingresando\_pin**.
- **Ingresando\_pin:** El usuario introduce su PIN. Si es correcto, se procede a seleccionar el tipo de transacción. Si el PIN es incorrecto, el sistema controla los intentos fallidos y puede emitir advertencias o bloquear el acceso tras tres intentos fallidos.
- **Tipo\_de\_trans:** Determina si la transacción es un depósito o un retiro.
- **Deposito:** Suma el monto ingresado al balance y actualiza la cuenta si el monto no es cero.
- **Retiro:** Verifica si el monto a retirar es menor o igual al balance. Si es suficiente, actualiza el balance y entrega el dinero; de lo contrario, se enciende la señal de "fondos insuficientes".
- **E\_Bloqueo:** Se activa después de tres intentos fallidos y solo puede resetearse con **rst**.

■ **Salidas:**

- **PIN\_INCORRECTO:** Se activa si el PIN ingresado es incorrecto.
- **ADVERTENCIA:** Se activa tras dos intentos fallidos.
- **BLOQUEO:** Se activa tras tres intentos fallidos.
- **ENTREGAR\_DINERO:** Indica que el cajero debe entregar el dinero en un retiro exitoso.

- **BALANCE\_ACTUALIZADO**: Señal que indica que el balance ha sido actualizado después de un depósito o retiro.
- **FONDOS\_INSUFICIENTES**: Señala que no hay suficientes fondos para realizar el retiro.

### 3. Plan de Pruebas

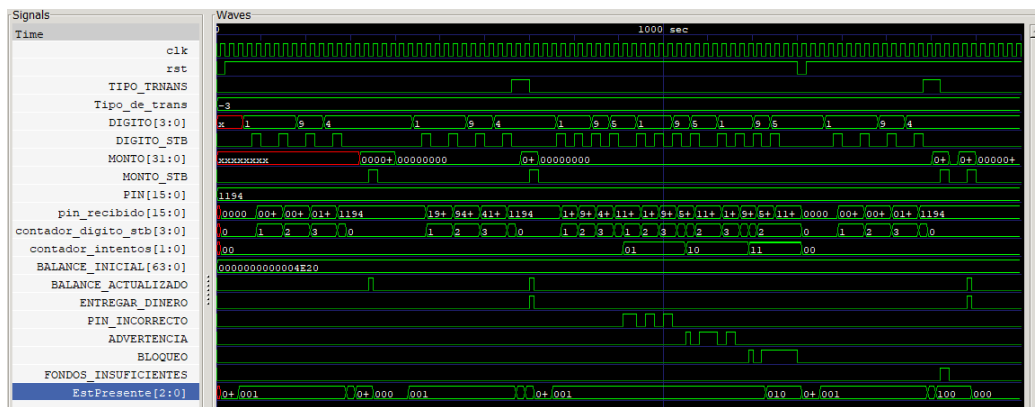


Figura 3: Plan de pruebas conductuales

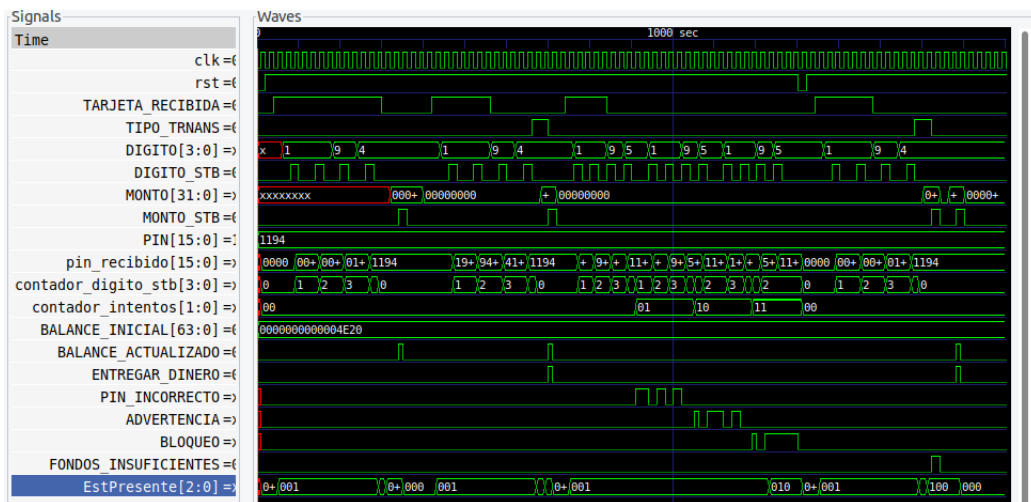


Figura 4: plan de pruebas estructurales

se puede ver que no hay mucha diferencia entre las pruebas conductuales y la prueba sintetizada. al parecer el balance desapareció del diagrama sintetizado además de los próximos contadores y el próximo estado de la máquina de cajero, sin embargo hay otra diferencia que es el tiempo de las salidas, al ser sintetizada hay mayor caso de que las salidas sean más angostas o salga tiempo después por el retardo

## 4. Resultados

Este plan de pruebas cubre las diferentes situaciones que puede enfrentar el controlador del cajero automático, desde transacciones exitosas hasta errores como intentos fallidos de PIN y retirados con fondos insuficientes. A continuación se detallan los casos de prueba.

## 5. Casos de Prueba

### 5.1. Caso 1: Depósito Perfecto

- **Descripción:** El usuario inserta una tarjeta y realiza un depósito exitoso.
- **Condiciones:**
  - Se inserta la tarjeta (`TARJETA_RECIBIDA = 1`).
  - Se ingresa correctamente el PIN (1194).
  - El tipo de transacción es depósito (`TIPO_TRNANS = 0`).
  - Se deposita un monto de 15,000.
- **Resultados esperados:** El balance se actualiza correctamente, la señal `BALANCE_ACTUALIZADO` se activa.

### 5.2. Caso 2: Retiro Perfecto

- **Descripción:** El usuario inserta una tarjeta y realiza un retiro exitoso.
- **Condiciones:**
  - Se inserta la tarjeta y se ingresa correctamente el PIN.
  - El tipo de transacción es retiro (`TIPO_TRNANS = 1`).
  - Se retira un monto de 15,000.

- **Resultados esperados:** El dinero es entregado (`ENTREGAR_DINERO = 1`), el balance se actualiza correctamente (`BALANCE_ACTUALIZADO = 1`).

### 5.3. Caso 3: PIN Incorrecto Tres Veces

- **Descripción:** El usuario inserta una tarjeta y falla en ingresar el PIN tres veces.
- **Condiciones:**
  - Se inserta la tarjeta y el usuario intenta ingresar el PIN incorrecto tres veces.
- **Resultados esperados:**
  - Después del primer y segundo intento fallido, se activa la señal `PIN_INCORRECTO`.
  - Después del tercer intento fallido, se activa la señal `BLOQUEO`, y el cajero queda bloqueado.

### 5.4. Caso 4: Retiro Fallido (Fondos Insuficientes)

- **Descripción:** El usuario intenta retirar más dinero del disponible.
- **Condiciones:**
  - El usuario intenta retirar 45,000, pero el balance es insuficiente.
- **Resultados esperados:** La señal `FONDOS_INSUFICIENTES` se activa y el retiro no se realiza.

## 6. Resumen de Comandos

A continuación, se presenta un resumen de los comandos disponibles en el `Makefile` para compilar, ejecutar y limpiar el proyecto:

- **Compilar, ejecutar y visualizar el diseño:**

```
1 make run
```

Este comando compila los archivos de Verilog, ejecuta la simulación y luego abre GTKWAVE para visualizar los resultados.



- **Limpiar los archivos generados:**

```
1 make clean
```

Elimina los archivos generados por la compilación y simulación, como `a.out`, `resultados.vcd`, y otros archivos intermedios.

- **Realizar la síntesis y ejecutar el diseño sintetizado:**

```
1 make synth
```

Ejecuta Yosys para sintetizar el diseño, simula el archivo sintetizado y abre GTKWAVE para visualizar los resultados.

## 7. Conclusiones y recomendaciones

al principio se tuvo problemas con el hecho del provador, las entradas era muy inmediatas y tuvimos que desplazarlas una a una para poder darle tiempo al retardo y que pudieran leerse bien las entradas.