

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE-0523 Circuitos Digitales II

1 ciclo 2023

Tarea #3

Máquina de estados:

Diseño de un controlador para cajero automático.

Daniel Alejandro Rodríguez Alvarado C06575

Grupo 001

Profesor: Enrique Coen Alfaro

7 de mayo del 2023

Índice

1. Resumen	1
2. Descripción Arquitectónica	2
3. Plan de Pruebas	4
4. Instrucciones de utilización de la simulación	4
5. Ejemplos de Resultados	5
6. Conclusiones y Recomendaciones	8

Índice de figuras

1.	Imagen decorativa. Panorama general de los resultados de onda obtenidos. . . .	1
2.	Descripción arquitectónica de la máquina de estados	3
3.	Primera etapa	5
4.	Segunda etapa	6
5.	Tercera etapa	7

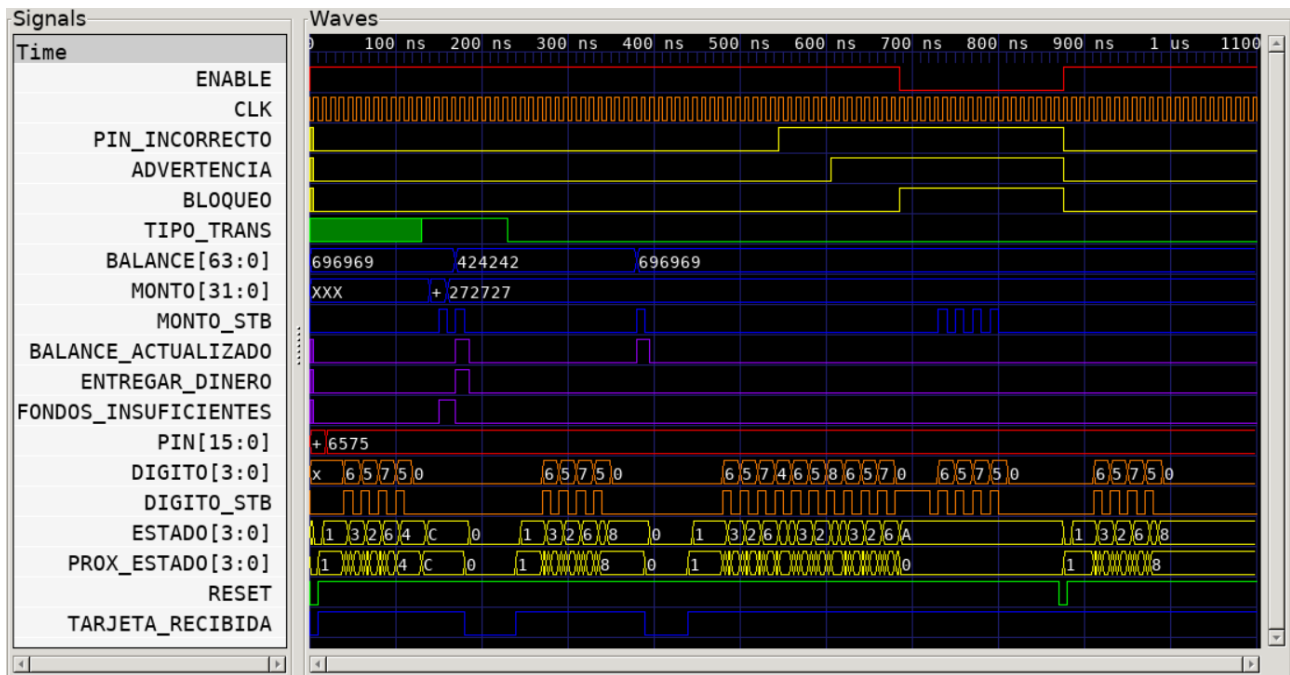


Figura 1: Imagen decorativa. Panorama general de los resultados de onda obtenidos.

1. Resumen

La tarea consiste en diseñar un controlador para un cajero automático que cumpla con ciertos requisitos de funcionamiento. Se debe implementar una máquina de estados que controle el flujo de operaciones del cajero, desde la recepción de la tarjeta y la entrada del PIN hasta la realización de operaciones de depósito o retiro de dinero.

La descripción conductual del controlador se realizará en Verilog, utilizando módulos de banco de pruebas y probador. Se hizo lo posible por que el plan de pruebas cubriese todos los modos de operación del cajero, incluyendo la detección de tarjeta insertada, entrada de PIN, comparación del PIN, detección de intentos fallidos de PIN y bloqueo del cajero, operaciones de depósito y retiro de dinero, verificación de fondos suficientes y actualización del balance de la cuenta.

Una característica que distingue este diseño en particular es que NO se utilizó un registro interno para llevar el conteo de intentos fallidos, sino que la activación de PIN_INCORRECTO habilita la posibilidad de activar ADVERTENCIA con el siguiente error, y así hasta llegar a BLOQUEO. Esta decisión no impacta negativamente en la funcionalidad del diseño, simplemente fue la manera que se me ocurrió en el momento.

En conclusión, el cajero automático diseñado muestra un correcto funcionamiento en cuanto a la detección y verificación del PIN, así como en la realización de transacciones de manera adecuada y segura. Además, la activación del bloqueo después del tercer intento fallido de ingreso de PIN demuestra una efectiva medida de seguridad para proteger la información del usuario.

2. Descripción Arquitectónica

El módulo cuenta con las partes estudiadas necesarias para una máquina de estados: definición o declaración de estados, definición de flip-flops, lógica de próximo estado y lógica de salidas. En cuanto a los estados utilizados se cuenta con los siguientes:

```
// Definición de estados
parameter ESPERA                = 4'b0000;
parameter INTRODUCIENDO_PIN_1  = 4'b0001;
parameter INTRODUCIENDO_PIN_2  = 4'b0011;
parameter INTRODUCIENDO_PIN_3  = 4'b0010;
parameter INTRODUCIENDO_PIN_4  = 4'b0110;
parameter TRANS_ELEGIR         = 4'b0100;
parameter TRANS_RETIRO         = 4'b1100;
parameter TRANS_DEPOSITO       = 4'b1000;
parameter PIN_INCORRECTO_1     = 4'b1001;
parameter PIN_INCORRECTO_2     = 4'b1011;
parameter PIN_INCORRECTO_3     = 4'b1010;
```

Cabe mencionar que el número de estados, a pesar de no ser muy grande, podría reducirse si se optara por utilizar salidas de Mealy, ya que la lógica de salidas implementada es casi en su totalidad salidas de Moore. Para entender un poco mejor la descripción arquitectónica del diseño, se armó el diagrama de la figura 2.

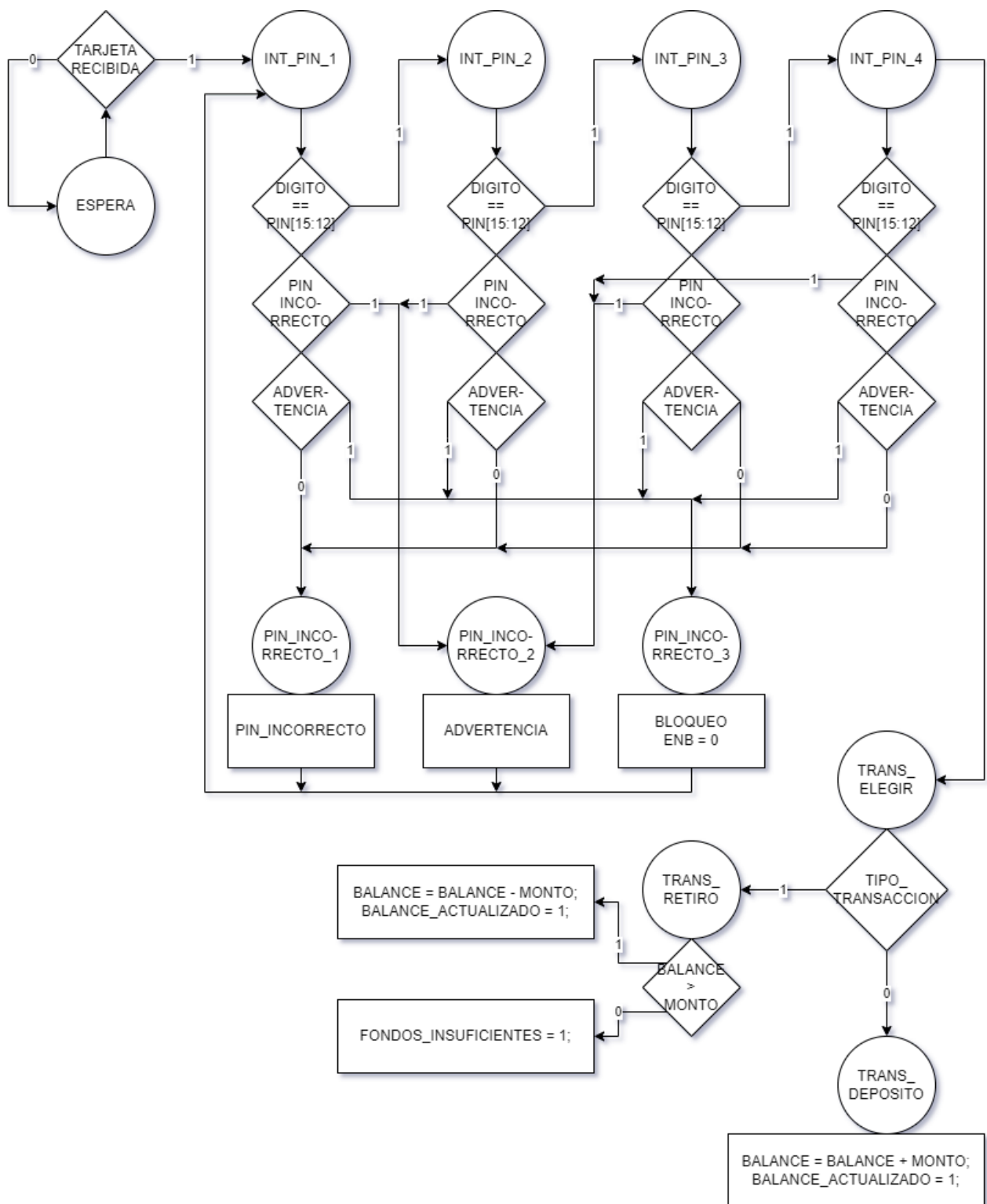


Figura 2: Descripción arquitectónica de la máquina de estados^{NOTA}

^{NOTA} Debido a la cantidad de estados, se excluyen algunos procedimientos para evitar un diagrama sobre-complicado.

3. Plan de Pruebas

Para la verificación funcional de la máquina de estados, se propone el siguiente plan de pruebas:

1. **Reset de registros:** Debe reiniciar los registros a un valor óptimo para iniciar el procedimiento normal de un cajero automático. También se utilizará posterior a que haya un BLOQUEO.
Evaluación: Exitoso ✓
2. **Recepción exitosa del PIN:** Una vez introducida la tarjeta (TARJETA_RECIBIDA activado), debe recibir progresivamente el PIN = 6575 (mi carné) mediante la entrada DIGITO y proceder con las etapas de transacción.
Evaluación: Exitoso ✓
3. **Etapas de transacción RETIRO:** De contar con el dinero, se retira el valor MONTO de BALANCE y dispara las señales ENTREGAR_DINERO y BALANCE_ACTUALIZADO, caso contrario, debe indicar FONDOS_INSUFICIENTES y permanecer en el estado de retiro sin disparar la señal de entrega. La acción se realiza con actividad en MONTO_STB. Ambos casos se toman en cuenta.
Evaluación: Exitoso ✓
4. **Etapas de transacción DEPÓSITO:** Añade el valor de MONTO a BALANCE. La acción se realiza con actividad en MONTO_STB y debe disparar la salida BALANCE_ACTUALIZADO.
Evaluación: Exitoso ✓
5. **Manejo de errores:** Tras haber introducido incorrectamente el PIN, cada intento fallido activa las siguientes señales en el siguiente orden: PIN_INCORRECTO → ADVERTENCIA → BLOQUEO.
Evaluación: Exitoso ✓
6. **Funcionalidad de BLOQUEO:** El estado diseñado para activar esta salida se encarga de deshabilitar un registro interno llamado ENABLE, lo cual permite frenar la acciones de la máquina. En esta prueba se activan diferentes posibles interacciones del usuario para demostrar que el BLOQUEO no permite hacer nada hasta que RESET vuelva a 1.
Evaluación: Exitoso ✓
7. **Retiro de Tarjeta:** Una vez se retira la tarjeta, TARJETA_RECIBIDA = 0, entonces la máquina vuelve a un estado de ESPERA, donde limpia registros internos y se prepara para recibir una tarjeta y empezar todo el proceso de nuevo. Se utiliza varias veces a lo largo del probador para separar etapas.
Evaluación: Exitoso ✓

4. Instrucciones de utilización de la simulación

El archivo makefile se encarga de compilar y correr la simulación. Únicamente debe correr lo siguiente en la línea de comandos:

```
make
```

5. Ejemplos de Resultados

Primera etapa

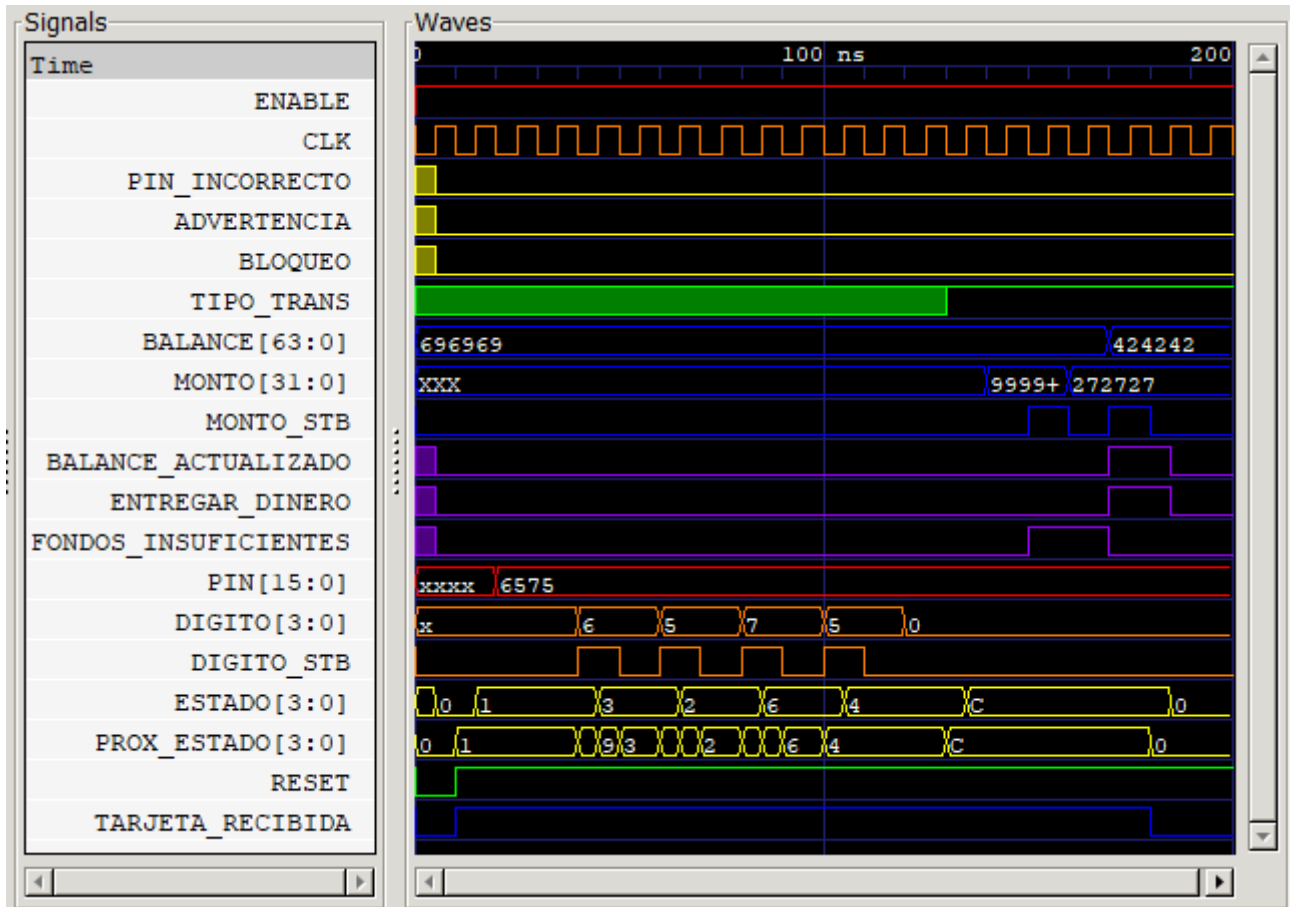


Figura 3: Primera etapa

La primera etapa a comentar, ver figura 3, corresponde a la de un ingreso exitoso del PIN, seguido de una transacción de retiro. Para esta tarea se decidió utilizar Pin = 6575, igual que mi carné. En la figura 3 se observa el ingreso progresivo del pin correcto. Cada dígito es tomado en cuenta en el momento que se activa el estímulo DIGITO_STB. Dado que el PIN termina de introducirse correctamente, entonces la máquina procede a un estado para la elección de transacción. Este estado intermedio sencillamente lee el valor de TIPO_TRANSACCION, en este caso indica que debe realizarse un retiro. El primer retiro NO es exitoso, porque se intentó retirar una cantidad de dinero MONTO superior a la que indica el BALANCE. A su vez, nótese la activación de la señal FONDOS_INSUFICIENTES. Posteriormente, se ajusta un valor de monto inferior al BALANCE y se logra apreciar como el balance en efecto reduce su valor y también se disparan las señales ENTREGAR_DINERO y BALANCE_ACTUALIZADO.

Esta primera etapa, igual que las demás, finalizan con el retiro de la tarjeta, es decir, TARJETA_RECIBIDA = 0. Esto reinicia exitosamente los registros internos al igual que las señales de transacción e intentos fallidos.

Segunda etapa

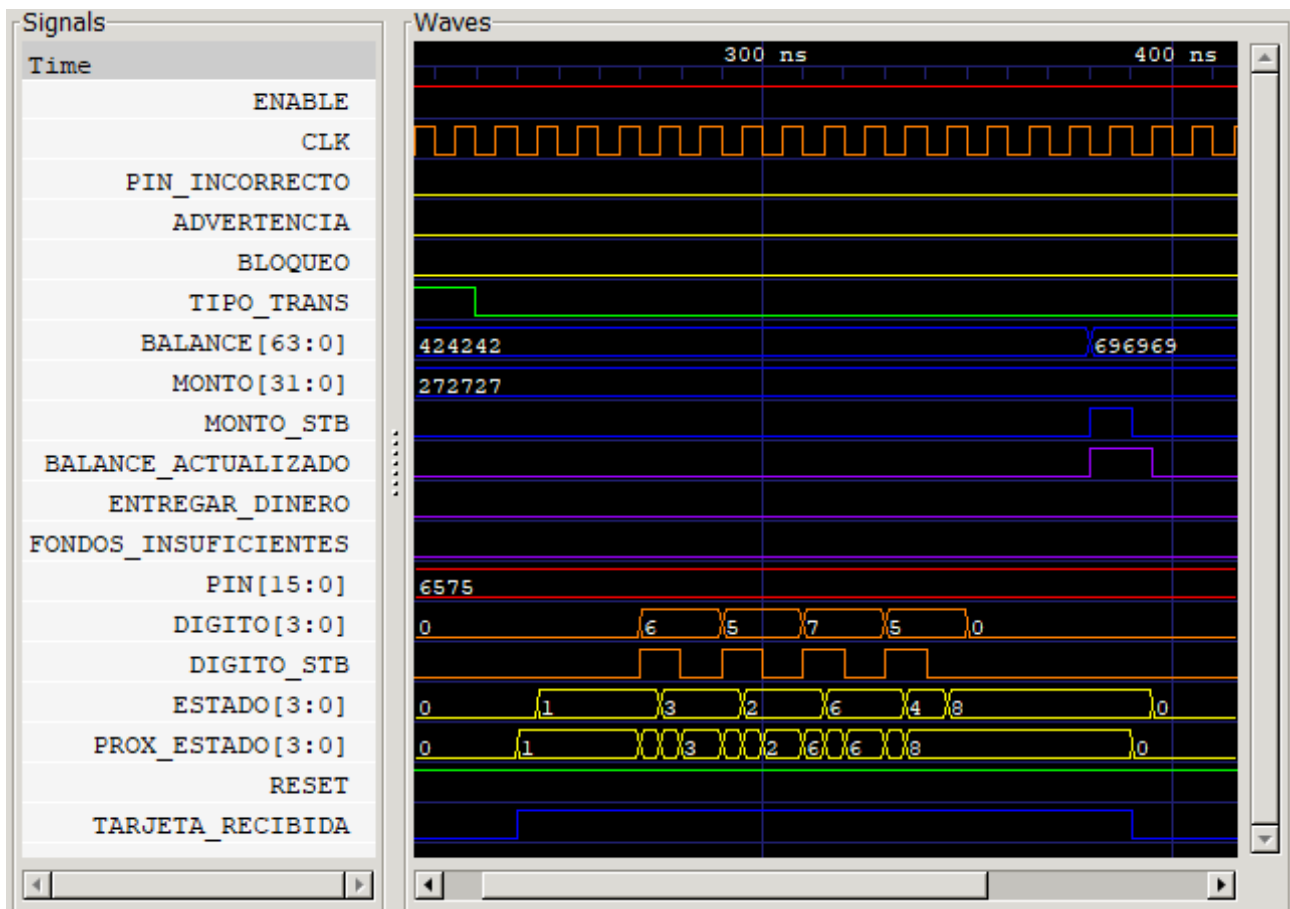


Figura 4: Segunda etapa

Con respecto a la segunda etapa, ver figura 4, esta es muy similar, cuenta nuevamente con la introducción correcta del PIN, con la excepción de que ahora TIPO_TRANS indica a la máquina qué debe realizar un depósito. Y en efecto, con la llegada de MONTO_STB, se observa como el valor indicado en MONTO se añade al BALANCE actual, disparando además la señal de BALANCE_ACTUALIZADO.

Tercera etapa

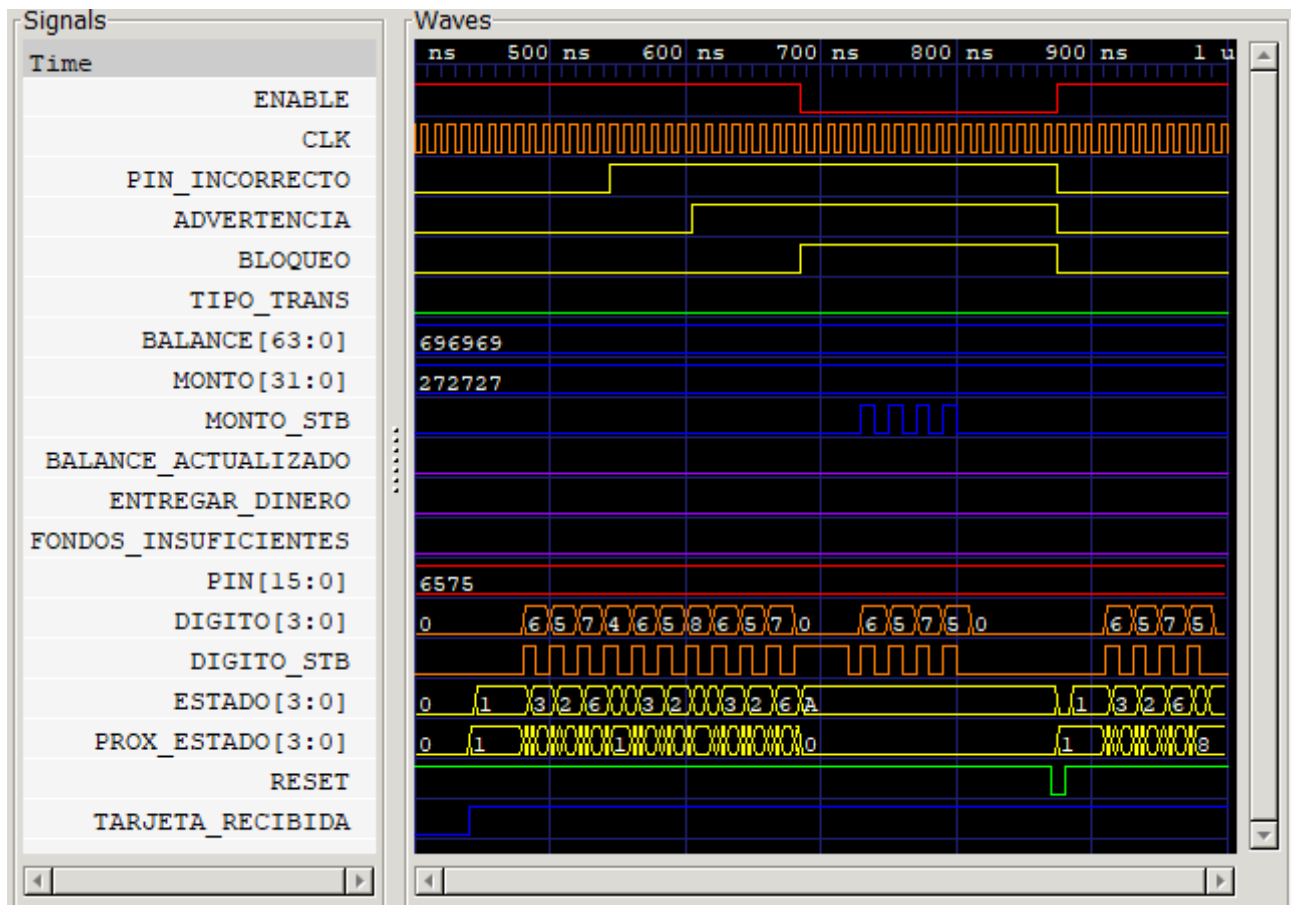


Figura 5: Tercera etapa

La etapa 3, ver figura 5, pretende fallar la introducción correcta del PIN hasta producir el bloqueo de la máquina. En la figura se muestran los dígitos introducidos, algunos correctos y otros incorrectos, de modo que permite apreciar qué incluso después de uno o dos intentos fallidos, todavía puede seguir con su normal funcionamiento. El bloqueo sucede sino hasta el tercer PIN incorrecto, llevando la máquina al estado `PIN_INCORRECTO_3` y disparando la salida `BLOQUEO`. Durante el bloqueo, la máquina deshabilita un registro interno llamado `ENABLE`, lo cual detiene el normal funcionamiento de la misma. La única manera de volver a habilitar este registro interno, es mediante la secuencia $0 \rightarrow 1$ de la señal `RESET`.

Obsérvese como inmediatamente después del flanco creciente del `RESET`, el estado presente de la máquina vuelve a `ESPERA` y vuelve exitosamente a su funcionamiento normal.

Adicionalmente, se añadió variedad de estímulos en las entradas para cuando el `BLOQUEO` estuviese activado, esto para demostrar la eficacia de dicho bloqueo.

6. Conclusiones y Recomendaciones

1. La implementación del ingreso del PIN es efectiva, permitió detectar de manera correcta los dígitos introducidos y realizar la verificación del PIN de manera exitosa.
2. Las transacciones como retiros y depósitos funcionaron de manera adecuada y solo cambiaban el valor del balance cuando era necesario, además de disparar las señales de salida necesarias para guiar al usuario.
3. Después del tercer intento fallido de ingreso de PIN, se activó el bloqueo respectivo de la máquina.
4. Bajo el estado PIN_INCORRECTO_3, no se permitió que estímulos de entradas a las que el usuario tendría acceso durante un bloqueo pudieran hacer algo, sino hasta después de la secuencia $0 \rightarrow 1$ de la señal RESET. Esto se ilustra en la figura 5.
5. La deshabilitación del registro interno ENABLE durante el bloqueo de la máquina es una medida efectiva para detener el funcionamiento de la misma y evitar posibles accesos no autorizados.
6. En fines prácticos, se sugiere implementar medidas adicionales de seguridad, como la detección de dispositivos fraudulentos o la activación de alarmas en caso de intentos de acceso no autorizados.
7. En cuanto a la lógica, se recomendaría añadir una entrada adicional TRANS_STB, para que la máquina utilice un estado intermedio en donde le pregunte al usuario qué tipo de transacción desea realizar, esto en lugar de asumir que el tipo de transacción es una entrada del módulo y ya, como se solicitaba en el enunciado.