

Reporte Tarea # 1

Brandon Jiménez Campos C33972

Resumen

En este reporte se presentará el diseño de un controlador de cajero automático, el cual realiza todas las funciones necesarias de un cajero, recibir una tarjeta, digitar el pin, realizar retiros, verificar que se puede entregar dinero o si no se cuenta con los fondos suficientes, además de que este controlador tiene la peculiaridad de que permite depósitos, función que muchos cajeros actualmente no contienen. Para verificar el correcto funcionamiento del diseño propuesto este pasó por varias pruebas, tales como, realizar un retiro, realizar un depósito, digitar el pin incorrecto, digitar el pin incorrecto y en el intento siguiente colocar el pin correcto y realizar un depósito, también realizar un usuario realizar un retiro y luego otro usuario realizar un deposito. Se logró poder diseñar correctamente un controlador de un cajero automático que permita realizar las funciones típicas, además de poder implementar esta novedad la cual permite generar depósitos, las pruebas demostraron que el controlador responde correctamente ante las situaciones planteadas para su debida prueba, por ende al responder correctamente se podría decir que este controlador está listo para poder implementarse en cajeros automáticos.

1. Descripción Arquitectónica

El diagrama de bloques utilizado para la creación de este diseño es el otorgado en el enunciado de la tarea al igual que la descripción de cada señal y cómo funciona. Por otro lado, el funcionamiento de la máquina de estados se encuentra en la figura 1, en la cual se detalla las condiciones para poder pasar de estados, el nombre de los estados y las salidas que se llegarán a activar en cada estado. A la hora de diseñar el controlador una de las partes que más dieron trabajo fue la lógica sobre cómo introducir los dígitos al pin ingresado, la solución a esto, fue colocarlos en la lógica secuencial utilizando asignaciones no bloqueantes, permitiendo así el poder colocar dígito por dígito. Además, cosas relevantes de este diseño es que en el código se colocó una parte de lógica combinacional, la cual trata solamente sobre las condiciones para poder pasar de estado, mientras que por otro lado se tiene una parte de lógica secuencial, la cual lo que realiza es todo lo que se debe de realizar en cada estado. También, cada vez que se entra al estado de elegir trámite y de esperando tarjeta se realiza una limpieza en los contadores, la señal de pin incorrecto, la de advertencia, entre otras. Esto con el fin de limpiar las señales cuando ya se colocó el pin correcto ó se está esperando una nueva tarjeta.

Nombre del estado	Número asignado al estado
Esperando Tarjeta	0
Digitar número	1
Se digitaron 4 números	2
Elegir trámite	3
Digitar monto retiro	4
Digitar monto depósito	5
Actualización balance	6
Actualización balance retiro	7
Pin incorrecto 1 vez	8
Pin incorrecto 2 veces	9
Pin incorrecto 3 veces	10

Cuadro 1: Estados y su número asignado en el código de verilog.

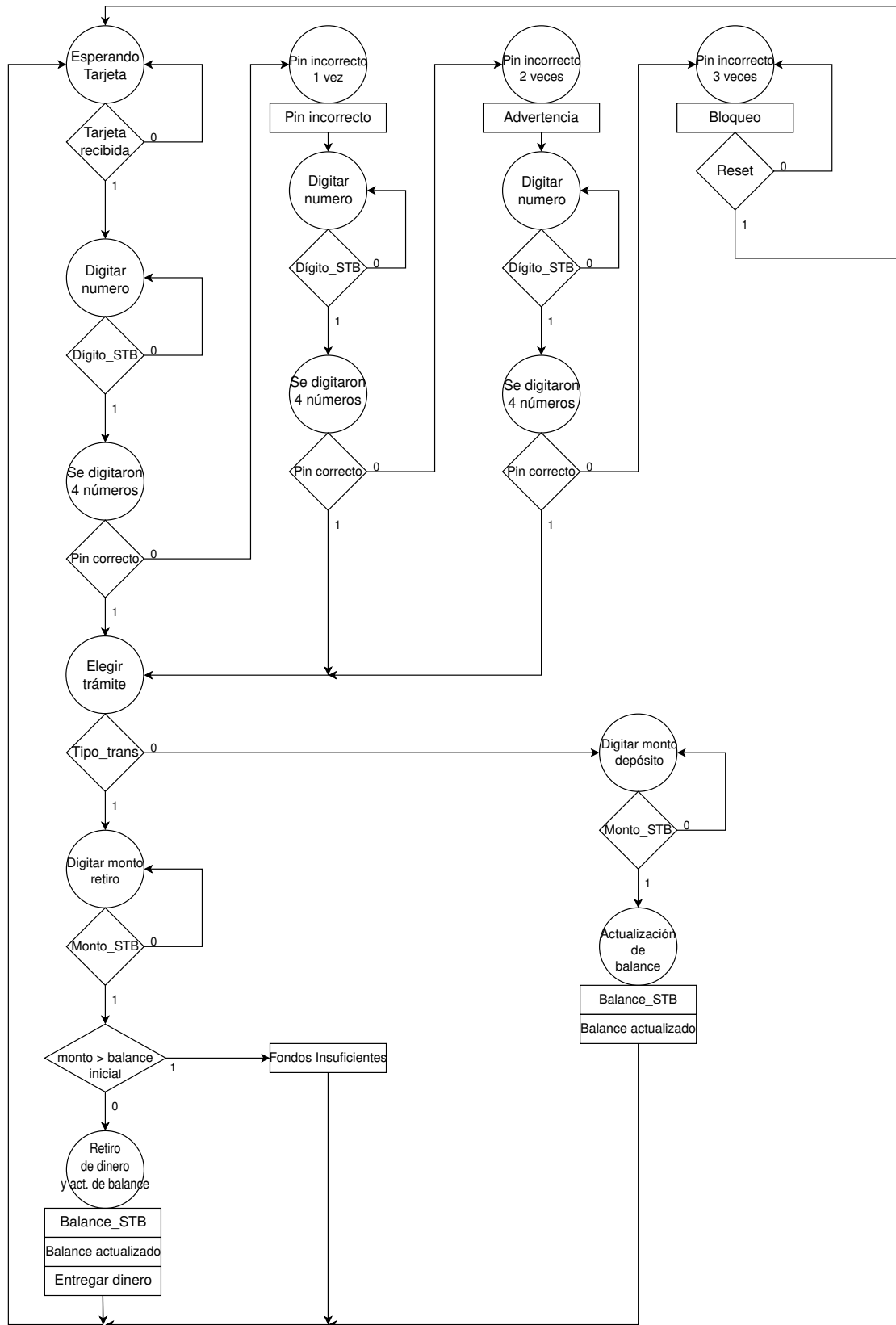


Figura 1: Diagrama ASM utilizado para el controlador del cajero automático.

2. Plan de Pruebas

El controlador fue puesto bajo las siguientes pruebas con el fin de poder verificar su correcto funcionamiento ante situaciones para las cuales está diseñado, como lo es un retiro o un depósito, tanto como situaciones que podría sufrir en el día a día, como podría ser el que una persona olvide su clave y falle el pin 3 veces o también que recuerde el pin en el segundo intento. A continuación se presentan las pruebas a las cuales fue sometido el diseño, además se planteó una situación muy común, la cual es que el usuario no disponga de los fondos suficientes, entonces también se realizó la prueba para observar cómo reaccionar el controlador en un caso como este.

Prueba I: Realización de un depósito básico.

Esta prueba consiste en una situación a la cual se va a enfrentar el controlador, ya que esta es una de sus funciones básicas, la cual es que el cajero recibe la tarjeta, el usuario coloca su pin correctamente, se selecciona el monto deseado para el depósito y se debe de actualizar el balance de la cuenta, esta prueba su finalidad es demostrar que el diseño cumple con lo solicitado correctamente.

Estado de la prueba: ✓ Aprobado.

Prueba II: Realización de un retiro básico.

Esta prueba consiste en una situación a la que se va a enfrentar el controlador de manera muy común, la cual es la de realizar un retiro, esta prueba consiste en que el cajero recibe una tarjeta, se ingresa el pin correcto y se selecciona el monto que se desea retirar y en este caso el monto va a ser menor que el balance inicial, logrando así que se actualice el balance y se active la salida de entregar dinero, esta prueba su finalidad es verificar que el diseño funciona correctamente y activa la señal de entregar el dinero.

Estado de la prueba: ✓ Aprobado.

Prueba III: Caso en donde la persona no tenga los fondos suficientes

Esta prueba consiste en una situación a la que muy probablemente vaya a enfrentar el controlador y esta es una prueba para verificar la correcta implementación de seguridad para que el cajero no otorgue más dinero del que contiene el usuario en su cuenta bancaria, ya que si el cajero otorga más dinero del que contiene el usuario sería un problema que perjudicaría al banco directamente.

Estado de la prueba: ✓ Aprobado.

Prueba IV: Se ingresa 3 veces un pin incorrecto

En esta prueba se verifica que el controlador sea capaz de poder actuar cuando un usuario falla el pin 3 veces, en este caso debería de bloquearse.

Estado de la prueba: ✓ Aprobado.

Prueba V: Un usuario realiza un trámite y luego llega otro usuario a realizar otro trámite

En esta prueba se verifica el funcionamiento del controlador cuando se tiene 2 trámites seguidos, esto ya que el controlador no solamente va a ser de un uso, este va a estar en un uso constante, por ende debe de funcionar correctamente ante dos trámites seguidos.

Estado de la prueba: ✓ Aprobado.

Prueba VI: Un usuario obtiene fondos insuficientes y luego otro usuario realiza un retiro

En esta prueba se verifica que el funcionamiento del controlador después de que a un usuario se le otorgó la señal de fondos insuficientes sea el correcto, este funcionamiento se verifica a través de cómo se comporta en la transacción posterior a que se activó la señal de fondos insuficientes.

Estado de la prueba: ✓ Aprobado.

Prueba VII: Un usuario bloquea el cajero y el siguiente usuario realiza un depósito.

A través de esta prueba se busca verificar el correcto funcionamiento del controlador en la transacción posterior a que se tuvo un bloqueo debido al introducir tres veces el pin incorrecto, luego de que se bloquee el controlador y se active la señal de reset para poder sacarlo de este estado de bloqueo y poder utilizarlo nuevamente se introducirá otra tarjeta en la cual se realizará un depósito.

Estado de la prueba: ✓ Aprobado.

Prueba VIII: Un usuario falla el pin 2 veces y en la tercera coloca correctamente el pin y el usuario que sigue también falla el pin 1 vez

Esta prueba consiste en que el primer usuario falla el pin 2 veces y en el tercer intento logra ingresar el pin correcto y realiza la transferencia que desea, seguido a esto el usuario que sigue también falla el pin 1 vez y en el segundo intento ya ingresa el pin correctamente y realizar el trámite deseado, esto es con el fin de poder observar que el controlador no acumule la cantidad de errores y se demuestre que cada vez que se ingresa una tarjeta el contador de fallos no arrastra números de usuarios anteriores.

Estado de la prueba: ✓ Aprobado.

Prueba IX: Usuario solicita todo el dinero que tiene en su cuenta

Esta prueba lo que realiza es verificar si el controlador otorga el dinero cuando el usuario coloca en monto su balance inicial, o sea $monto = balance_inicial$, y verificar si realiza la actualización del balance y activa el $balance_STB$.

Estado de la prueba: ✓ Aprobado.

3. Instrucciones de utilización de la simulación

Para poder ejecutar la simulación se presentará a continuación la manera de poder ejecutarlo desde la terminal en linux. Para esto primero se debe de descargar los archivos nombrados como controlador.v, tester.v, testbench.v y Makefile, esto debido a que se necesitan los 4 archivos para poder ejecutar la simulación, porque en el controlador.v esta el código del controlador, en tester.v está toda la información de valores que toman las señales que entran al controlador y el tiempo en donde se activan las señales, el testbench.v que es donde se conecta al controlador con el tester y por último el archivo Makefile que será el encargado de realizar la simulación. Para este proyecto se creó un makefile para facilitar la ejecución de la simulación, por lo que para poder utilizar la simulación y poder observar las señales en el visualizador de formas de ondas GTKWave, por lo que en el directorio (carpeta) en donde se ubican los archivos mencionados anteriormente se coloca el comando make, de manera que se observe como en el listado 1

Listado 1: Colocación del comando make

```
user@ubuntu:~$ make
```

Cuando se desee eliminar los archivos generados por el make se utiliza el comando mostrado en el listado 2

Listado 2: Comando para limpieza de archivos generados por el make.

```
user@ubuntu:~$ make clear
```

Si no se desea utilizar el make, en el listado 3 se presenta el código que se debe de utilizar.

Listado 3: Comandos para ejecutar simulación

```
user@ubuntu:~$ iverilog testbench.v
user@ubuntu:~$ vvp a.out
user@ubuntu:~$ gtkwave resultados.vcd
```

4. Ejemplos de resultados

Los resultados obtenidos se presentarán a través de cómo se comportó el controlador ante algunas pruebas de las mencionadas en el plan de pruebas, en este reporte se mostrarán solamente algunas de las pruebas más relevantes.

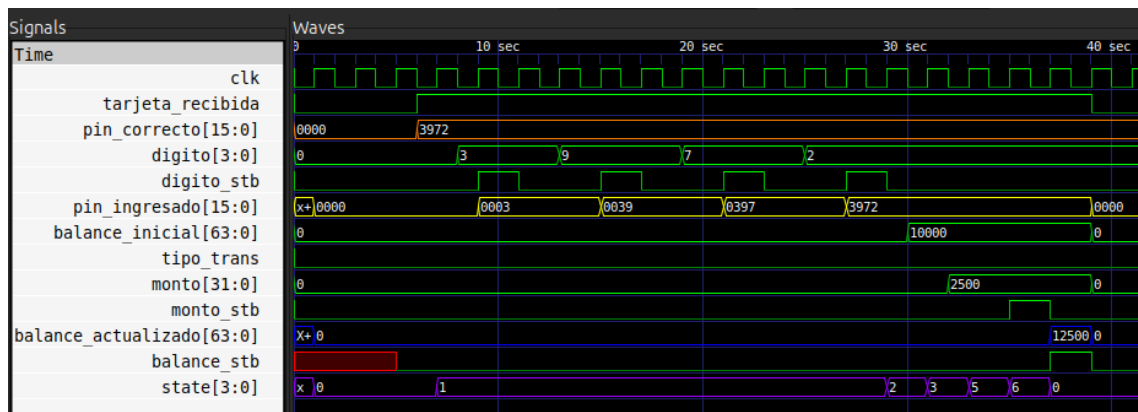


Figura 2: Resultados al realizar la prueba I.

En la figura 2, se puede observar cómo cuando se enciende la señal de que se recibió una tarjeta, inmediatamente aparece el pin correcto (la señal naranja), además de que luego de que se colocan los 4 dígitos, la señal pin ingresado (la amarilla) es igual a la de pin correcto, por ende se pasa de estado, esto se puede observar en la señal de state (la morada), la cuál es la señal en el estado que se encuentra, los cuales para saber en que estado se encuentran se pueden observar en la tabla 1, al ver que *tipo_trans* es cero, se trata de un depósito, por lo que el balance actualizado sería la suma de monto y de balance inicial y efectivamente, en la señal azul se observa que el balance actualizado es la suma del balance inicial y del monto depositado.

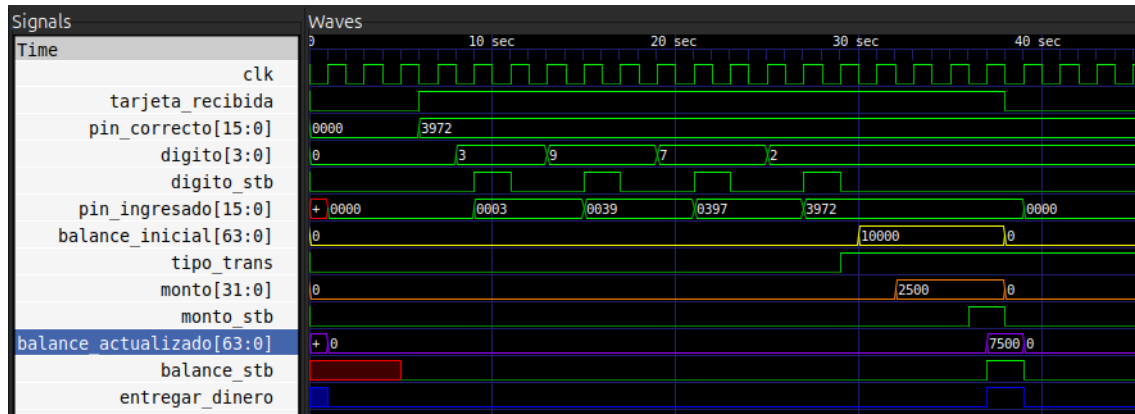


Figura 3: Resultados al realizar la prueba II.

En la figura 3, la señal amarilla representa el balance inicial del usuario, como *tipo_trans*, está en 1 se desea realizar un retiro, por lo que el monto digitado por el usuario debe ser menor a balance inicial, en la señal naranja, la cual corresponde al monto, se observa que el monto es 2500 y este valor claramente es menor al balance inicial, el cual es de 10000, por lo que el balance actualizado seria la resta de estos $10000 - 2500 = 7500$ y seguidamente de que el balance se actualizó se enciende la señal de *balance_stb* por un ciclo de reloj, al igual que la salida de entregar dinero.

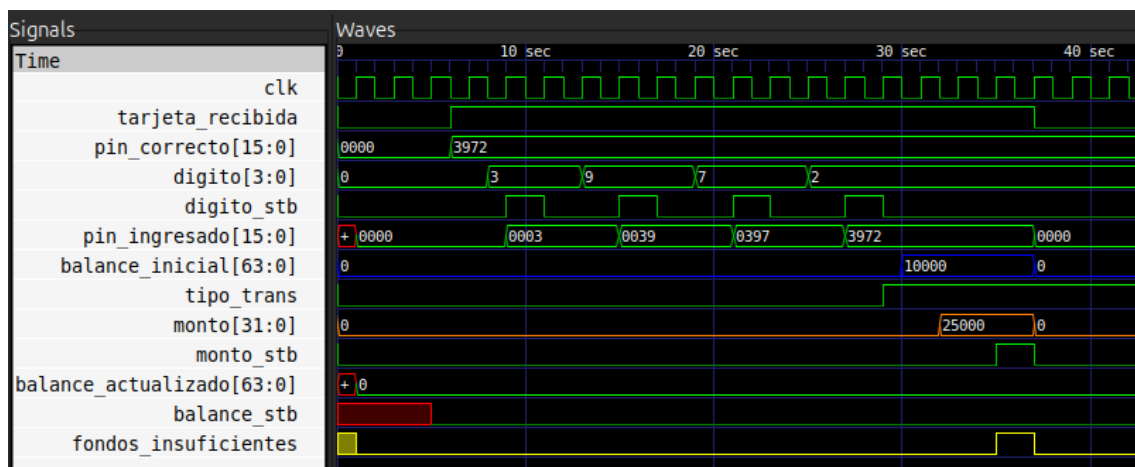


Figura 4: Resultados al realizar la prueba III.

En la figura 4, la señal azul representa el balance inicial del usuario, en este caso, al igual que el de la figura 3 el *tipo_trans* está en 1, por lo que se desea realizar un retiro, en este caso se puede observar a través de la señal naranja que el monto ingresado es 25 000, mayor al balance inicial, por lo que a través de la señal amarilla se puede observar que se enciende la salida de fondos insuficientes, además que también se observa como el balance no se actualiza y *balance_stb* tampoco se activa.

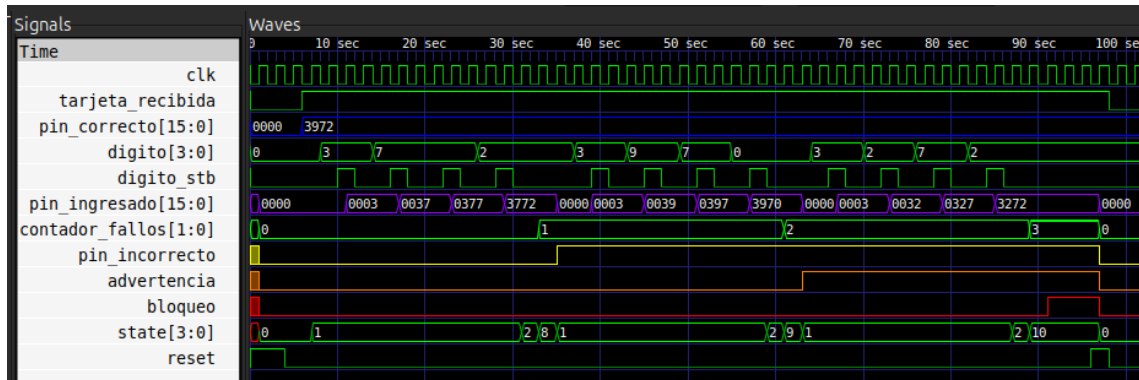


Figura 5: Resultados al realizar la prueba IV.

En la figura 5, se observa en la señal azul que al ingresar la tarjeta, el pin correcto es 3972, sin embargo, en la señal morada se observa que los pines ingresados por el usuario no son el correcto, es por esto que se comienzan a activar ciertas salidas, por ejemplo, la señal amarilla es la salida de pin incorrecto, que se activa cuando se falla por primera vez el pin, la señal naranja es la de advertencia, esta se activa cuando se falla por segunda vez el pin y por último la señal roja es la de bloqueo, esta es la que se activa cuando se falla 3 veces el pin y para poder salir de este estado de bloqueo el cajero se debe de reiniciar, la cual es la señal de reset, presentada debajo de la de bloqueo y se observa que luego de activarla se vuelve al estado cero, el cual es el estado inicial de esperando tarjeta, tal y como se mencionó en la descripción arquitectónica, en la tabla 1

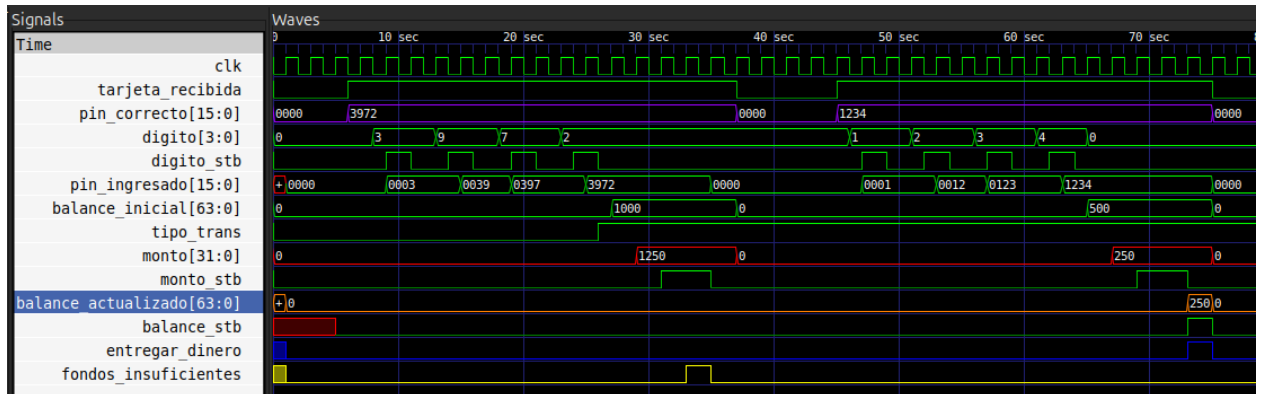


Figura 6: Resultados al realizar la prueba VII.

En la figura 6, se puede observar a través de la señal morada, que se tienen dos pines correctos, esto hace referencia a que son dos usuarios distintos, en donde ambos desean realizar un retiro, el primero a través de la señal roja se observa que el monto que desea retirar es mayor a su balance inicial, por lo que la salida de fondos insuficientes se activa, esto se observa en la señal amarilla y su balance no se actualiza, mientras que el segundo usuario sí logra digitar un monto que le permite realizar un retiro válido, por lo que se observa en la señal naranja que su balance se actualiza y en la señal azul se activa la salida de entregar dinero, mientras que en su trámite la señal de fondos insuficientes no se activó.

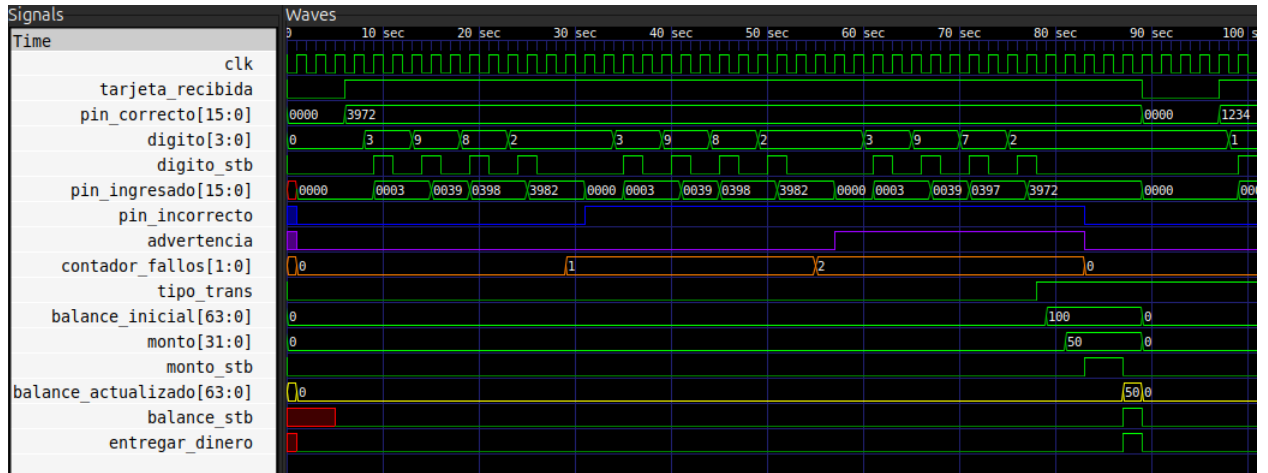


Figura 7: Resultados prueba VIII y prueba IX (parte 1).

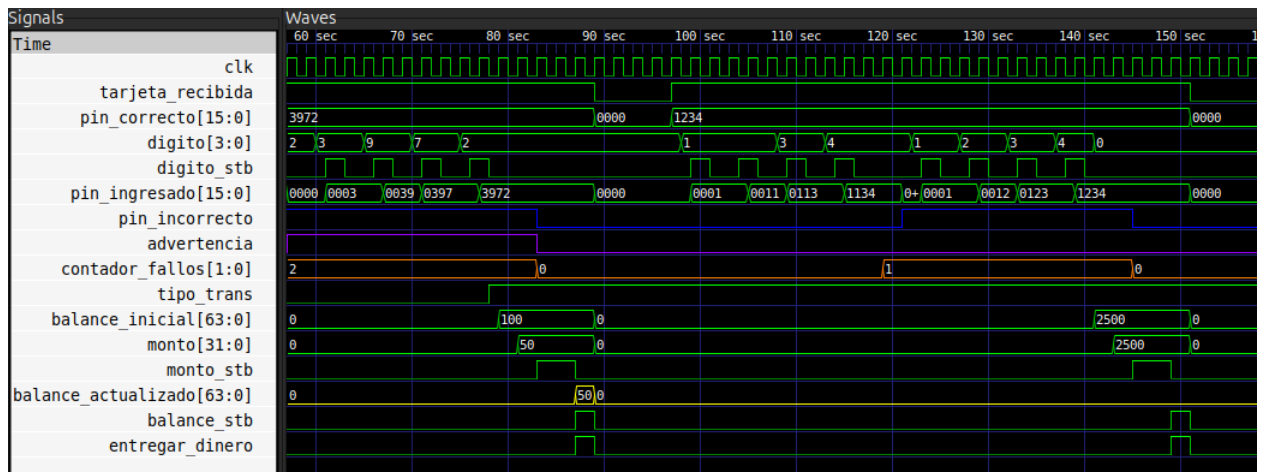


Figura 8: Resultados prueba VIII y prueba IX (parte 2).

Por último se presenta en las figuras 7 y 8 los resultados que se obtuvieron al realizar las pruebas VIII y prueba IX juntas, empezando en la figura 7 se observa en donde el primer usuario falla el pin dos veces, se puede ver como la salida de pin incorrecto se activa (la señal azul) y la salida de advertencia (la señal morada) también se activa, así como en la señal naranja se puede observar el contador de fallos, también en la señal amarilla se observa el balance actualizado y al mismo tiempo se activan las señales de *balance_STB* y la de entregar dinero, ahora en la figura 8 se observa inicialmente cómo la señal naranja no arrastra el contador obtenido anteriormente, sino que está en cero, por lo que cuando el segundo usuario falla el pin por primera vez, el contador no lo toma como una tercera vez y no se bloquea, además la señal de balance actualizado, la señal amarilla, no toma ningún valor, pero esto es porque el balance actualizado al tener $\text{monto} = \text{balance_inicial}$ el balance actualizado será cero, se puede ver que *balance_STB* y entregar dinero sí se activan, por ende si está funcionando.

Las demás pruebas se pueden derivar de estas principales, además en el archivo *tester.v* se encuentra una adicional que no se colocó en los ejemplos de resultados

5. Conclusiones y recomendaciones

A través de los resultados obtenidos se observa que se logró diseñar de manera correcta un controlador para un cajero automático que permita realizar retiros y depósitos, además de contar con funcionalidades de seguridad, como lo es verificar que el pin ingresado es el correcto o es inválido, además de verificar si se cuenta con los fondos necesarios para poder realizar el retiro es una funcionalidad necesaria para garantizar un correcto funcionamiento y garantizar que el banco no se vea perjudicado. Con este diseño propuesto es importante tener en cuenta la importancia que se le debe de otorgar a la velocidad del reloj, ya que el diseño al estar basado en comportamientos los cuales se va a necesitar utilizar flip flops D estos funcionan con el flanco positivo del reloj, entonces si el reloj es demasiado lerdo el usuario podría observar que el cajero no avanza a las funcionalidades y esto se debe a que la máquina de estados avanza de estado es en un flanco positivo de reloj, entonces la frecuencia del reloj es bastante importante y algo a tomar en cuenta para poder implementar este controlador. Además esto se ve reflejado en uno de los problemas que se presentaron, ya que durante la creación del archivo `tester.v`, muchas veces el pin no se ingresaba correctamente, faltaban números y esto ocurrió debido a que no se sincronizaba muy bien el `digito_STB` con los ciclos del reloj.

El proyecto se recomienda que se planee con un tiempo prudente, ya que a lo largo de las pruebas surgen errores o malos funcionamientos y una buena forma de encontrar errores y resolverlos es durante las pruebas, pero para esto se debe de realizar con un tiempo prudente, además de que con un lapso de tiempo grande se puede lograr una gran cantidad de pruebas y proponer un mejor plan de pruebas.