

UNIVERSIDAD DE COSTA RICA

Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0523 - Circuitos Integrados Digitales
II ciclo de 2024

Tarea #3: *Controlador para Cajero Automático*

Profesor:
Enrique Coen Alfaro

Estudiante:
Gabriel Siles Chaves - C17530

Grupo 01

6 de octubre 2024

Índice

1. Resumen	2
2. Descripción Arquitectónica	2
2.1. Síntesis	3
3. Plan de Pruebas	4
3.1. Prueba #1. Retiro ingresando monto mayor al balance y reingresando monto	5
3.2. Prueba #2. Ingreso de pin incorrecto 3 veces	5
3.3. Prueba #3. PIN incorrecto 2 veces y se realiza depósito	5
4. Instrucciones de utilización de la simulación	6
4.1. Linux	6
5. Resultados	6
5.1. Resultado Prueba #1 Diseño Conductual	7
5.2. Resultado Prueba #1 Diseño Estructural	7
5.3. Resultado Prueba #2 Diseño Conductual	8
5.4. Resultado Prueba #2 Diseño Estructural	8
5.5. Resultado Prueba #3 Diseño Conductual	9
5.6. Resultado Prueba #2 Diseño Estructural	9
6. Conclusiones y recomendaciones	10

1. Resumen

En esta Tarea #3 se diseña una maquina de estados para un controlador de cajero automática que responde a entradas y opera correspondientemente a las especificaciones del enunciado. Se realiza tanto un diseño conductual el cual se estructura con un total de 8 estados que generan salidas en función a las entradas, una vez realizado el diseño por medio de la herramienta de *yosys* se realiza una síntesis completa de la descripción conductual para posteriormente obtener la descripción estructural para poder verificar el funcionamiento correcto del controlador. El sistema tiene un funcionamiento básico, recibe una tarjeta que contiene un balance inicial y pin por lo que el controlador se encarga de realizar el manejo de fondos dependiendo lo que el usuario desee realizar, teniendo en cuenta que bloquea el sistema en caso que el pin se incorreo. Al igual, el controlador es capaz de indicar si el monto que se desea retirar excede del balance que contiene la tarjeta.

En este reporte se describe todo el proceso y el funcionamiento del controlador, detallando paso por paso la estructura, pruebas, instrucciones de ejecución y los resultados obtenidos con sus respectivas conclusiones.

2. Descripción Arquitectónica

En este caso a la hora de realizar un controlador de cajero automático se emplea una diseño de máquina de estados que contenga estados y transiciones con respecto a las entradas para así activar las salidas. La máquina de estados que se puede observar en la Figura 1 tiene un total de 8 estados los cuales son los siguientes:

- **a Estado inicial, Lectura de tarjeta:** Se espera que el cajero reciba una tarjeta para iniciar el sistema, este estado es el inicial.
- **b Ingreso Dígito:** Se ingresa un dígito por lo que dependiendo a lo que se ingresa se guardará en una variable interna llamada corroborar un 1 en N.
- **c Comprobación de PIN:** EN este estado se comprueba si el pin ingresado es el correcto, si corroborar es 1111 será correcto de lo contrario incorrecto, teniendo en consideración que se tiene 3 intentos por lo que se lleva un contador máximo de 3.
- **d Tipo de transacción:** En este estado se recibe la opción que el usuario desea ingresar, retiro (1) o depósito (0). No podrá generar una decisión hasta el momento que el usuario ingrese tipo_stb.
- **e Bloqueo:** En el momento que se llega al tercer intento incorrecto el sistema se bloquea por lo que hasta que el reset sea 0 se puede desbloquear.
- **f Depósito:** El usuario ingresa la opción de depositar dinero por lo que ingresa un monto y este se ingresa al balance.
- **g Retiro:** El usuario ingresa la opción de retirar dinero por lo que si el monto excede el valor de balance inicial este indicará que los fondos son insuficientes, en caso contrario se entrega el dinero.

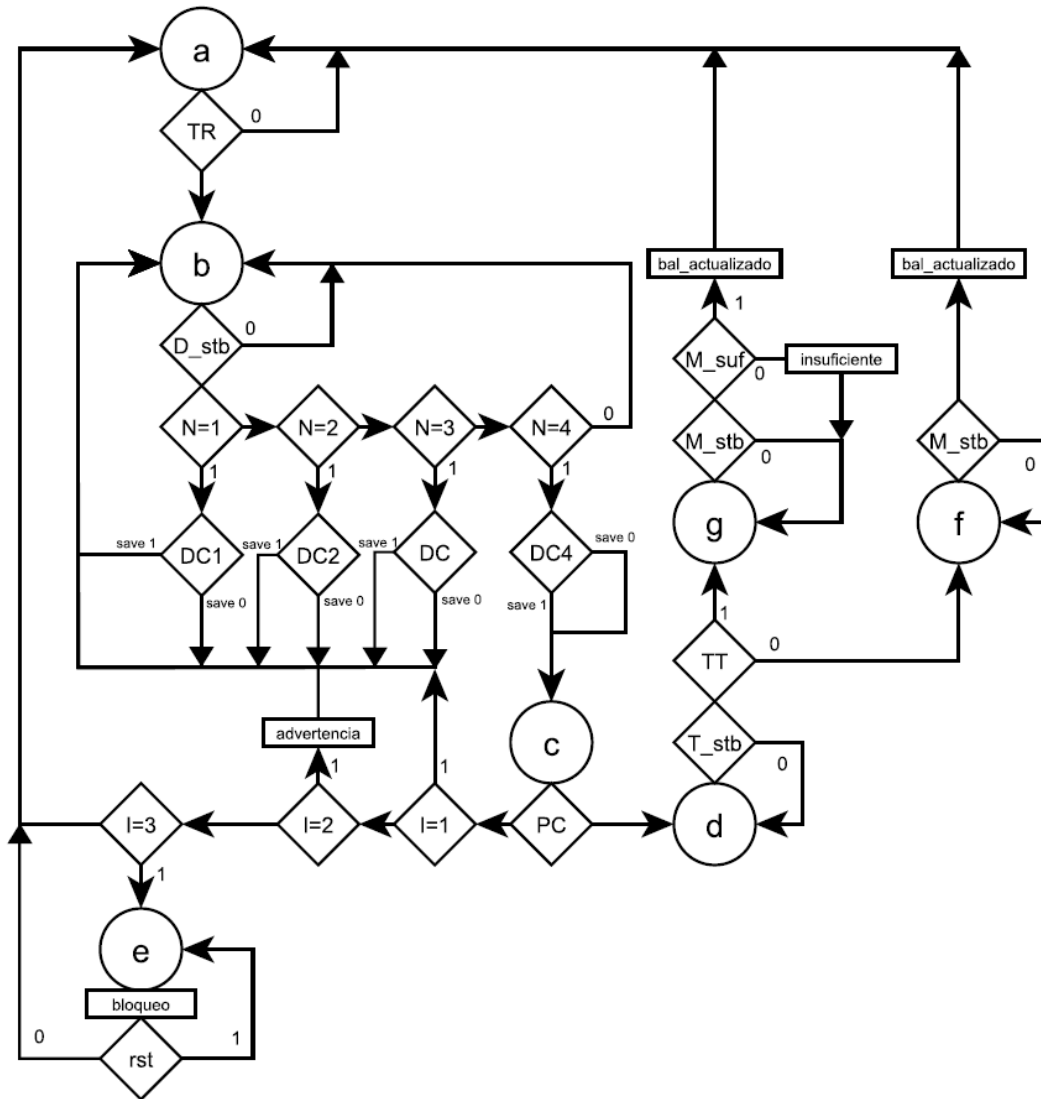


Figura 1: Diagrama ASM máquinas de estado cajero automático

Es importante tener en consideración el funcionamiento de la variable corroborar y el contador N de ingreso. A la hora de que se ingresa un dígito el contador N aumenta en uno y este realiza una corroboración de dígito. La variable de corroborar tiene 4 bits, en el momento de que un dígito es correcto en corroborar[4-N] se guarda un 1, en caso contrario será un 0. Por lo que el valor de corroborar debe ser 4'b1111 para que el pin sea correcto. Por último se agregó una entrada llamada tipo_stb, es importante a la hora de seleccionar una opción entre retirar o depositar debido a que el sistema de ser capaz de entender el momento que el usuario ingresó el valor para pasar de estado.

2.1. Síntesis

A la hora de realizar la síntesis del diseño conductual para convertirlo en un diseño estructural la herramienta de *yosys* ejecuto varias líneas de comando las cuales se puede observar en la Figura 2 en donde se utiliza la librería de *cmos_cells.lib*.

```

1  # read design
2  read_verilog cajero_automatico.v
3
4  # elaborate design hierarchy
5  hierarchy -check -top cajero
6
7  # the high-level stuff
8  proc; opt;
9
10 # mapping to internal cell library
11 techmap; opt
12
13 # mapping flip-flops to cmos_cells.lib
14 dfflibmap -liberty ./cmos_cells.lib
15
16 # mapping logic to cmos_cells.lib
17 abc -liberty ./cmos_cells.lib
18
19 #Generate GraphViz output
20 show
21
22 # cleanup
23 clean
24
25 # write synthesized design
26 write_verilog cajero_synth.v
27

```

Figura 2: Diagrama ASM máquinas de estado cajero automático

Al ejecutar este archivo .ys se genero los siguiente componentes:

Componente	Cantidad
NAND	301
NOT	296
NOR	126
DFF	19
Señales internas	507
Entradas	141
Salidas	25

Tabla 1: Componentes del diseño

3. Plan de Pruebas

En esta sección se presenta la lista detallada de las pruebas de cada estado y corroborar el funcionamiento del sistema de acuerdo a las especificaciones.

3.1. Prueba #1. Retiro ingresando monto mayor al balance y reingresando monto

Se realiza prueba en la situación en donde el usuario ingresa la tarjeta, donde el cajero obtiene el pin y el balance inicial. Continuando con el usuario ingresando 4 dígitos que son 2302, ingresando un pin correcto. El usuario ingresa que desea retirar e ingresa un valor mayor al balance inicial, por lo que debe volver a ingresar el monto esta vez correcto actualizando el balance y entregando el dinero. Al realizar la prueba el **diseño pasó la prueba**.

- **01:** Se ingresa tarjeta (PIN = 2302 y Balance Inicial = 50 331 648)
- **02:** Se ingresa 2302 como PIN
- **03:** Se ingresa 1 como tipo de transacción (retiro)
- **04:** Se ingresa monto de 83 886 080
- **05:** Se reingresa un monto de 5 242 880

3.2. Prueba #2. Ingreso de pin incorrecto 3 veces

Esta segunda prueba simula la situación en donde el usuario ingresa la tarjeta e ingresa incorrectamente 3 veces el PIN por lo que el sistema al primer intento indica pin incorrecto, a lo que vuelve a esperar el ingreso del PIN, al segundo intento el sistema indica de nuevo pin incorrecto y tira una advertencia y al tercer intento el sistema se bloquea. Al bloquearse la única manera para salir de este estado es apagar el reset, por lo que se desactiva y se vuelve a activar para reiniciar el sistema.

- **01:** Se ingresa tarjeta (PIN = 2302)
- **02:** Se ingresa 1234 como PIN
- **03:** Se ingresa 2352 como PIN
- **04:** Se ingresa 2309 como PIN
- **05:** Se desactiva reset (rst=0)
- **05:** Se activa reset (rst=1)

3.3. Prueba #3. PIN incorrecto 2 veces y se realiza depósito

En la tercer el usuario se equivoca al ingresar el PIN 2 veces por lo que al observar la advertencia revisa el PIN en su celular e ingresa al tercer intento el PIN correctamente por lo que decide depositar un monto al balance de la tarjeta realizando el proceso correctamente.

- **01:** Se ingresa tarjeta (PIN = 2302 y Balance Inicial = 45 088 768)
- **02:** Se ingresa 1234 como PIN
- **03:** Se ingresa 2352 como PIN

- **04:** Se ingresa 2302 como PIN
- **05:** Se ingresa 0 como tipo de transacción (depósito)
- **06:** Se ingresa monto de 2 097 152

4. Instrucciones de utilización de la simulación

Para ejecutar los comandos necesarios para realizar la simulación se debe contar con unos requisitos previos para la utilización del mismo, se debe tener instalado las siguientes aplicaciones:

- [Icarus Verilog](#)
- [GTKWave](#)
- [Yosys](#)

4.1. Linux

En caso de que se desee ejecutar en alguna extensión de Linux (Ubuntu, Debian, Xubuntu) se debe correr una terminal en la carpeta donde se encuentran todos los archivos necesarios para la ejecución de cada parte.

Para ejecutar el diseño conductual con sus respectivas ondas se debe ingresar el siguiente comando:

```
>> make conductual
```

Para ejecutar la síntesis para realizar el archivo para el diseño estructural se debe correr en la terminal el siguiente comando:

```
>> make sintesis
```

Una vez realizado el proceso de síntesis se puede realizar la descripción estructural y observar las ondas, es importante que primero se realice el proceso de síntesis y después se ejecute el siguiente comando:

```
>> make estructural
```

5. Resultados

A la hora de ejecutar el testbench se genera un archivo llamado *results.vcd* el cual al abrirse con el GTKWave se podrá visualizar las ondas y comprobar el funcionamiento del sistema.

5.1. Resultado Prueba #1 Diseño Conductual

Al aplicar el tester al *cajero_automatiko.v* se obtiene una serie de resultados, las salidas que se activan o cambian con respecto a las entradas. En la Figura 3 se visualiza el funcionamiento correcto del la primera prueba.

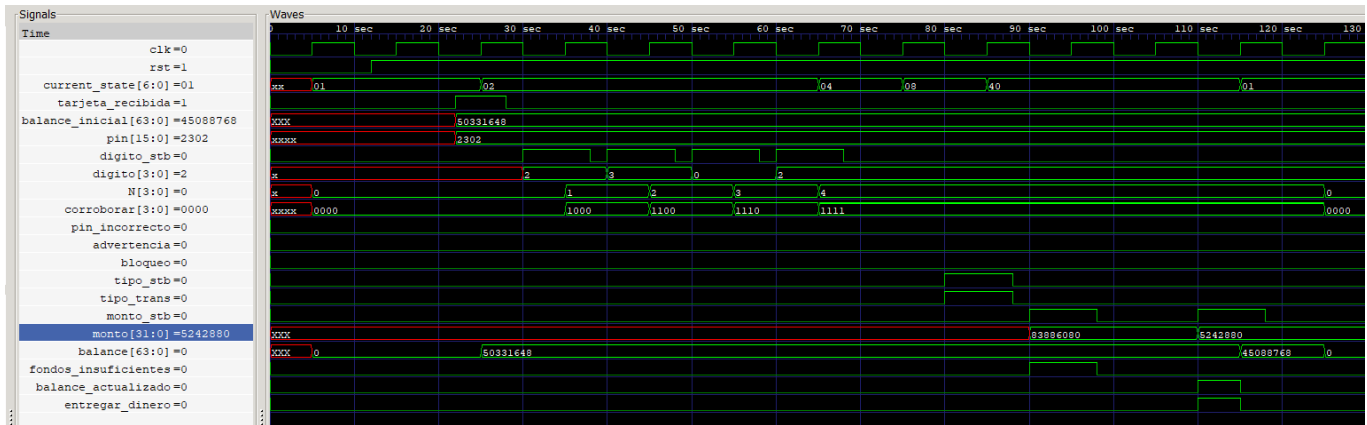


Figura 3: Ondas del GTKwave Conductual Prueba #1

Se puede observar como la variable interna corroborar va guarda 1s cuando el dígito es correcto, al igual que el contador aumenta cuando se ingresa el dígito. De forma correcta la salida de fondos insuficientes se activa y se genera correctamente el retiro cuando se ingresa el monto, se puede observar que el valor del balance disminuye.

5.2. Resultado Prueba #1 Diseño Estructural

Para el caso del diseño estructural se puede observar en la Figura 4 las diferentes entradas y salidas. Es importante saber que en este caso no se puede observar el balance que es una variable interna debido a que al realizar el proceso de síntesis esta variable como es interna no se realiza un acción específica como tal por lo que no se incluye en los resultados, en caso que se deseara observar se podría crear un balance output que permita observar el balance.

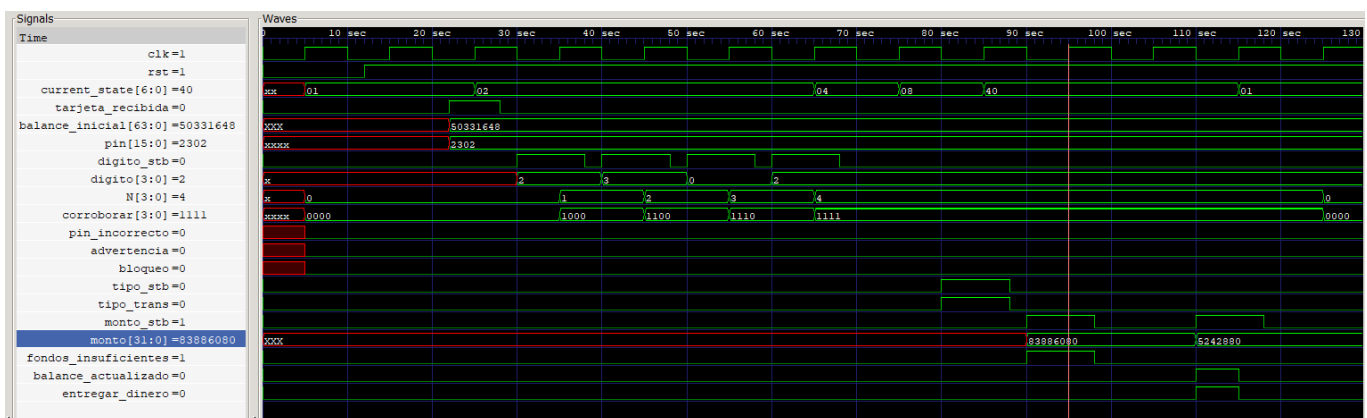


Figura 4: Ondas del GTKwave Estructural Prueba #1

5.3. Resultado Prueba #2 Diseño Conductual

Para la segunda prueba se obtiene un resultado esperado en donde se puede observar como al ingresar incorrectamente el PIN aumenta el contador de intentos, además se activan las salidas de advertencia, bloqueo y pin?incorrecto en el momento que se comprueba que el pin es incorrecto. En este caso es importante observar como la variable corroborar va guardando 0 cuando el dígito es incorrecto.

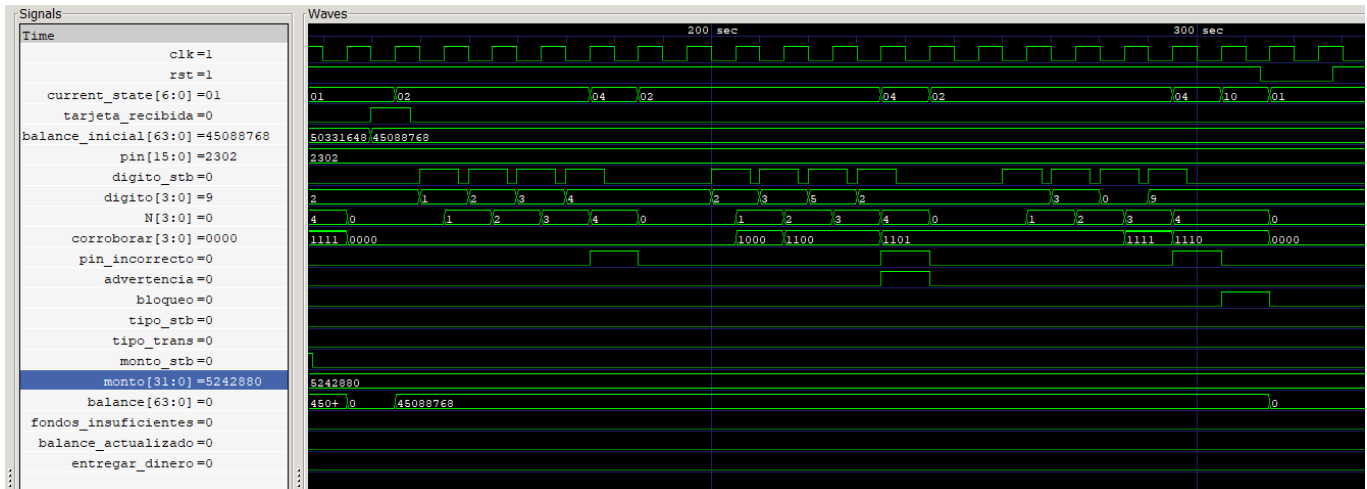


Figura 5: Ondas del GTKwave Conductual Prueba #2

5.4. Resultado Prueba #2 Diseño Estructural

En la Figura 6 las salidas trabajan de manera correcta y de igual forma a la esperada, se indica cuando el pin es incorrecto y cuando se activa la advertencia y se genera el bloqueo. Se puede notar que al momento de realizar el reinicio se desbloquea el sistema. Es notorio como tanto el diseño conductual como el estructural tienen el mismo comportamiento, únicamente que no se observa el balance como tal en las ondas del diseño estructural.

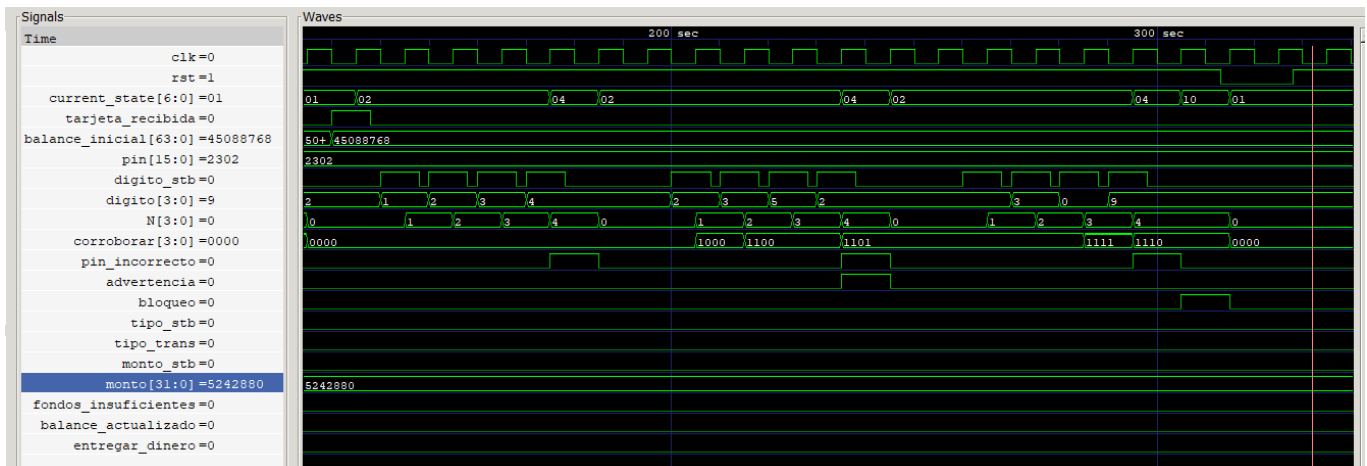


Figura 6: Ondas del GTKwave Estructural Prueba #2

5.5. Resultado Prueba #3 Diseño Conductual

Para la tercera prueba se puede visualizar en la Figura 7 las ondas como funcionan correctamente, a la hora de ingresar el PIN correcto se comprueba que está bien y permite seleccionar el tipo de transacción que se desea realizar, por lo que al retirar 2 097152 se obtiene un balance de 47185920.

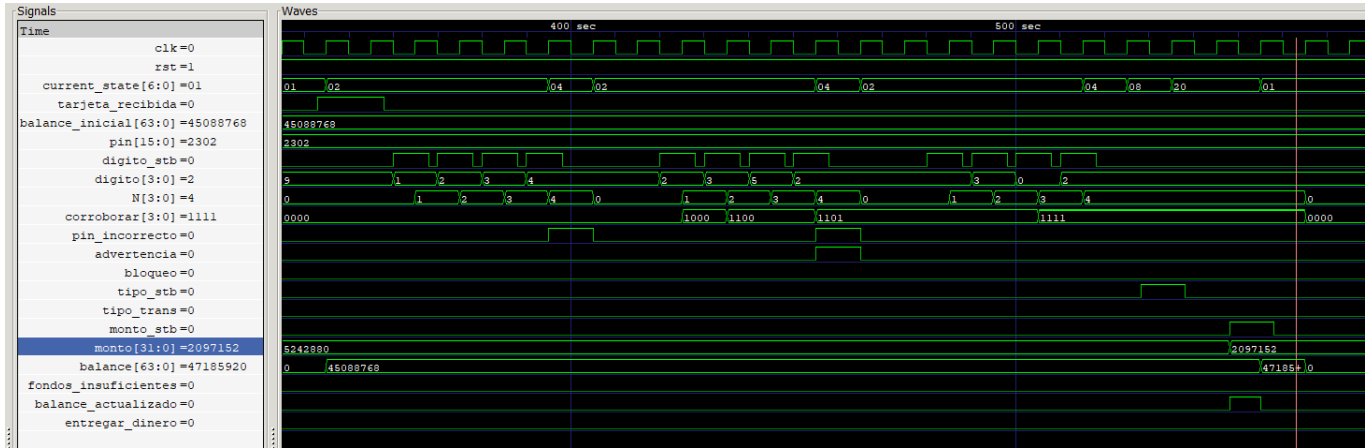


Figura 7: Ondas del GTKwave Conductual Prueba #3

5.6. Resultado Prueba #2 Diseño Estructural

Para el último resultado se puede observar en la Figura 8 un comportamiento correcto en donde la selección del usuario del tipo de transferencia es respetada por el sistema y permite realizar el depósito, actualizando el balance interno.

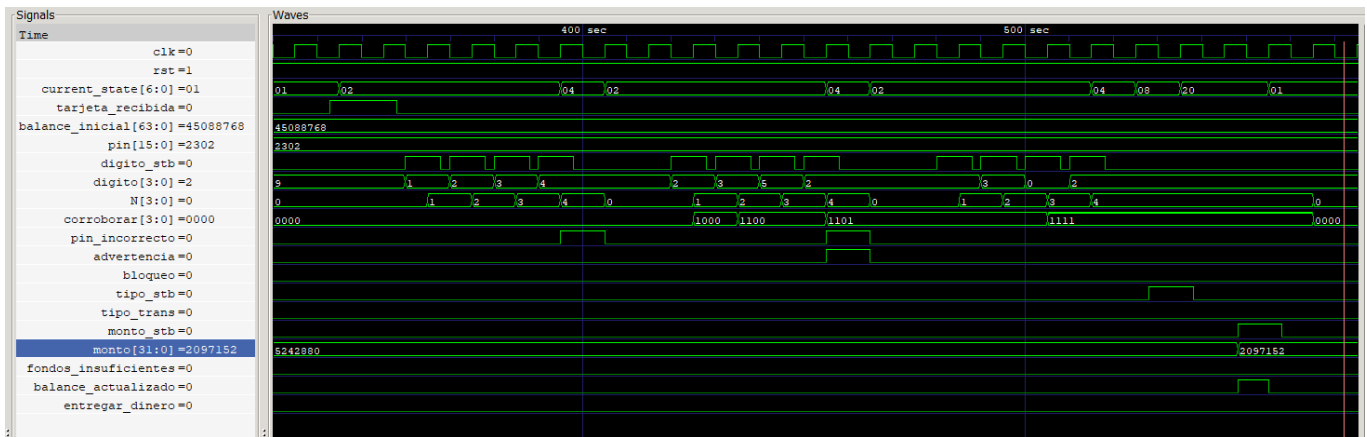


Figura 8: Ondas del GTKwave Estructural Prueba #3

6. Conclusiones y recomendaciones

En conclusión, el diseño conductual del cajero automático en comparación con el controlador de estacionamiento que se realizó en la Tarea #1 conlleva mucho más complejidad, teniendo aproximadamente el doble de estados. Es importante que se definan entradas que pueden facilitar el diseño completo del sistema debido a que hubieron fallos a la hora de realizar ciertas condiciones como por ejemplo en el tipo de transacción, al no indicar cuando se ingresaba la selección el sistema generaba una lectura incorrecta de la opción escogida por el usuario.

En este caso al realizar la síntesis se tuvo que evitar ciertos pasos como el *fsm* y *memory* debido a que generaba que no se mappearan los wires ni los gates generando un diseño estructural incorrecto y vació el cual no permitía el funcionamiento del controlador. En general en la parte de generar el diseño estructural se tuvo más problemas que se arreglaron pero que a futuro se pueden implementar cambios como para poder observar el balance interno en la parte estructural.

Es recomendable como siempre tener cuidado con los tiempos en que se ingresan las entradas debido a que pueden generar lecturas incorrectas, por lo que el *tester.v* es de suma importancia y requiere un diseño cuidadoso para ajustarse al clock y permitir que se lea de manera correcta en los flancos. En general se pudo realizar el objetivo de la tarea de manera eficiente y correcta cumpliendo con las especificaciones de las entradas y salidas.