

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

Curso IE-0523: Circuitos Digitales II

II ciclo 2023

Tarea 3

Leonardo Leiva Vasquez C14172

Grupo 01

Profesor: Enrique Coen Alfaro

10 de octubre, 2023

1. Resumen

El presente trabajo se basa en el diseño de un controlador para un cajero automático que aborda diversas operaciones, tales como la recepción de tarjetas, verificación de PIN, depósitos, retiros y actualización de balance, además de gestionar errores. Este diseño se detalla conductualmente en Verilog para un análisis claro de su funcionamiento. El proyecto incorpora un módulo de pruebas, un módulo probador y un módulo principal descriptivo. Contamos con un exhaustivo plan de pruebas que abarca todas las funcionalidades del cajero. Se realizó la síntesis del controlador con Yosys y la biblioteca cmoscells.lib, con toda la documentación pertinente. Finalmente, se efectuó una verificación post-síntesis para asegurar la adecuada operación del netlist sintetizado. En el tester.v, se puso a prueba nuestro sistema con situaciones variadas, desde el ingreso repetido de PIN incorrecto hasta retiros con fondos insuficientes. Si bien en general el controlador demostró un comportamiento robusto y fiable, se detectaron inconvenientes en la síntesis usando ciertas máquinas, lo que requirió ajustes y verificaciones adicionales. Como conclusión, este proyecto reafirma la importancia de una detallada simulación y prueba en cualquier diseño de controlador, en especial cuando se trata de operaciones financieras. Para futuros diseños, se sugiere considerar la incorporación de medidas de seguridad adicionales y tal vez explorar la adaptabilidad del controlador a diferentes plataformas bancarias y tecnologías emergentes.

2. Descripción Arquitectónica

La arquitectura del controlador para nuestro cajero automático se basa en una máquina de estados que gestiona las principales operaciones, como recepción de tarjetas, verificación de PIN y manejo de transacciones.

2.1. Diseño de la micro-arquitectura

A continuación, presentamos el diagrama de estados que ilustra el proceso de construcción de la micro-arquitectura del cajero automático. Esta representación gráfica permite entender de manera clara y concisa las transiciones y estados clave que componen el sistema, facilitando su análisis y comprensión.

1. **Estado ESPERA_TARJETA (7'b0000001):** En este estado, el controlador aguarda la inserción de una tarjeta por parte del usuario. Es la fase inicial donde el cajero está listo para iniciar una nueva transacción.
2. **Estado INGRESO_PIN (7'b0000010):** Una vez detectada la tarjeta, el sistema entra en este estado, solicitando al usuario que ingrese su PIN personal. Es el paso esencial para autenticar al titular de la cuenta.
3. **Estado COMPROBAR_PIN (7'b0000100):** En este punto, el controlador verifica la autenticidad del PIN proporcionado. Compara el número ingresado con el almacenado en la base de datos para confirmar o denegar el acceso.
4. **Estado SELECCION_TRANSACCION (7'b0001000):** Con la autenticación exitosa, el sistema permite al usuario seleccionar el tipo de transacción que desea realizar, ya sea un depósito, un retiro o consultar el balance.
5. **Estado BLOQUEO (7'b0010000):** Si el usuario introduce un PIN incorrecto en tres ocasiones consecutivas, el controlador entra en este estado. Bloquea temporalmente el acceso, garantizando la seguridad de la cuenta ante intentos de acceso no autorizados.

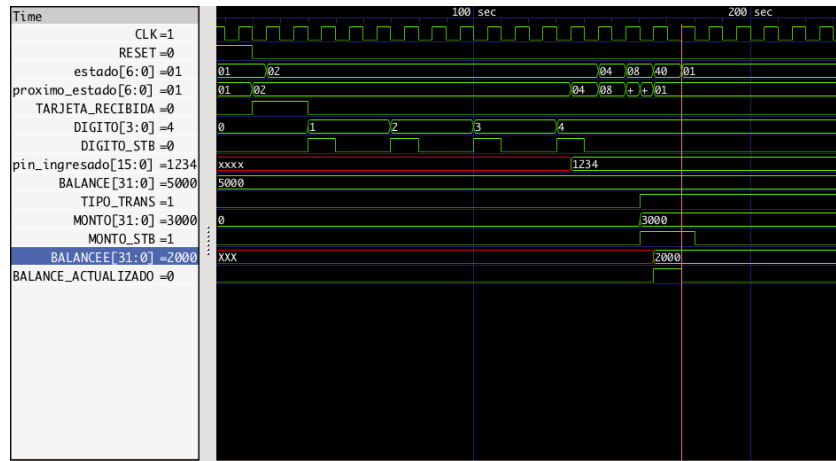


Figura 2: Retiro

2. **Retiro con fondos insuficientes** Esta prueba se centra en el escenario en el que el usuario intenta realizar un retiro sin contar con los fondos necesarios. El diagrama que presentamos a continuación, generado a partir del archivo .vcd, ilustra las señales activadas en este caso en particular y nos permite entender las acciones y respuestas del cajero automático ante tal situación.

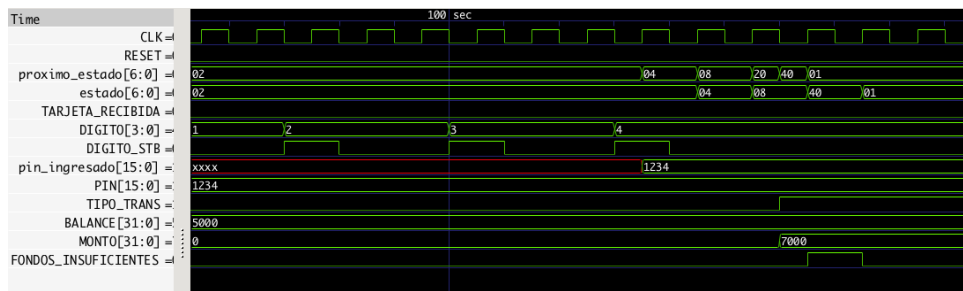


Figura 3: Fondos insuficientes

3. **Deposito** En esta instancia, el enfoque es el proceso de depósito de dinero en el cajero automático. La siguiente visualización, derivada del archivo .vcd, destaca las señales que se activan durante la fase de depósito, ofreciendo una perspectiva más profunda de las interacciones y mecanismos involucrados en esta operación.

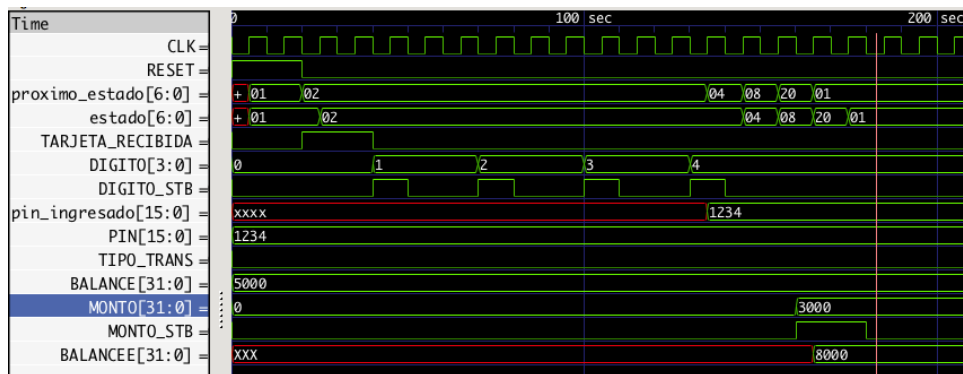


Figura 4: Deposito

4. **Se ingresa pin incorrecto tres veces** En esta fase de pruebas, observamos cómo el sistema responde al ingreso repetido de un PIN incorrecto. Es esencial notar las señales

correspondientes a las alertas y el eventual bloqueo del sistema tras la tercera entrada incorrecta. La figura que mostramos a continuación proporciona una representación visual de estas interacciones y respuestas del sistema.

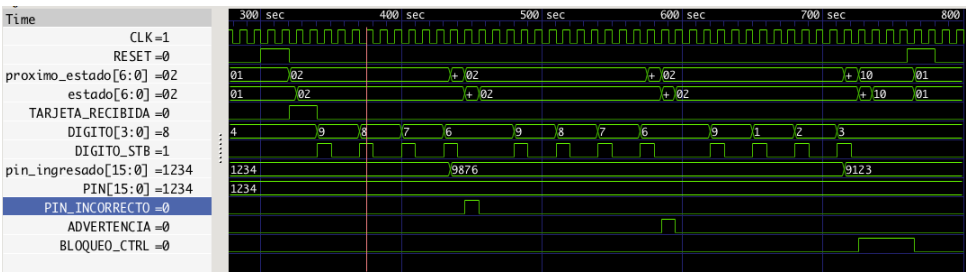


Figura 5: Bloqueo al ingresar pin incorrecto por tercera vez

5. **Pin err3neo 3 veces en el caso del controlador sintetizado** Esta prueba verifica el comportamiento del controlador al introducir tres veces un PIN incorrecto. En el primer intento err3neo, el controlador activar3 la se3al "PIN incorrecto". Al segundo error, se esperar3a la activaci3n de la se3al de "advertencia". Finalmente, con un tercer PIN err3neo, el controlador emitir3 la se3al de "bloqueo", previniendo cualquier acceso adicional. Esta funcionalidad es crucial para mantener la seguridad del sistema frente a intentos no autorizados.

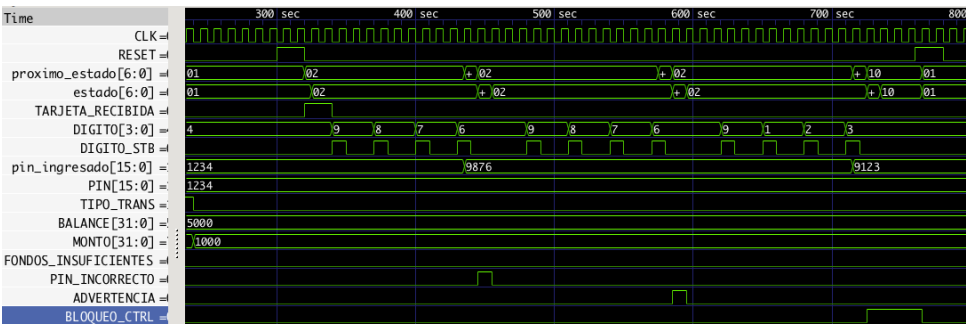


Figura 6: Bloqueo al ingresar pin incorrecto por tercera vez con el controlador sintetizado

Numero de compuertas y Flip Flops empleados

A continuaci3n, se presentan im3genes que detallan la cantidad de compuertas y Flip Flops utilizados en el dise3o del controlador.

12.1.2. Re-integrating ABC results.		
ABC RESULTS:	NOT cells:	16
ABC RESULTS:	NAND cells:	26
ABC RESULTS:	NOR cells:	42
ABC RESULTS:	_const0_ cells:	1
ABC RESULTS:	internal signals:	174
ABC RESULTS:	input signals:	39
ABC RESULTS:	output signals:	10
Removing target directory		

Figura 7: Compuertas utilizadas

mapped 6 \$_DFF_P_ cells to \DFF cells.

Figura 8: Flip Flops utilizados

4. Instrucciones de utilización de la simulación

El proceso de ejecución del controlador es notablemente sencillo: se lleva a cabo utilizando el comando `make`. Es importante mencionar que, en la configuración del `MAKEFILE`, se optó por utilizar el diseño original (`.v`) en lugar del diseño sintetizado. Esta elección no fue arbitraria. Durante el proceso de desarrollo, me enfrenté a una situación peculiar: en una computadora, la síntesis producía un error, mientras que en otra máquina la síntesis se llevaba a cabo sin inconvenientes. A pesar de este obstáculo, quiero enfatizar que la síntesis fue exitosa y, para respaldar esta afirmación, he incluido pruebas concretas en otra sección del reporte. Mi decisión de enfocarme en el diseño `.v` para el `Makefile` se basó en garantizar la consistencia y evitar posibles incongruencias debido a variaciones en el entorno de hardware. En términos generales, puedo afirmar con confianza que el programa funciona al y cumple con todas las especificaciones establecidas.

5. Conclusiones y recomendaciones

A lo largo de este proyecto, se desarrolló un controlador integral para un cajero automático empleando una máquina de estados. Esta metodología permitió una gestión eficiente de las distintas operaciones del cajero, desde la recepción de tarjetas hasta la ejecución de transacciones. A pesar de los desafíos encontrados, especialmente en la síntesis del controlador, el resultado final fue satisfactorio y se alinea con las expectativas iniciales.