Simulación de una máquina de estado en Verilog

Brenda Romero Solano (B96953); Alejandra Solano Ramírez (C17621); Jun Hyun Yeom Song (B17326)
Ingeniería Eléctrica, Universidad de Costa Rica

I. Introducción

Los circuitos lógicos se clasifican en dos tipos: combinacionales y secuenciales. Las salidas de los circuitos lógicos combinacionales sólo dependen de sus entradas actuales, mientras que las salidas de los circuitos lógicos secuenciales también pueden depender de entradas recibidas en tiempos pasados arbitrarios. En la práctica, casi todos los circuitos lógicos son *secuenciales*. [1]

El estado de un circuito lógico secuencial se puede definir como un conjunto de variables de estado cuyos valores en cualquier tiempo determinado contienen toda la información del pasado necesaria para explicar el comportamiento futuro del circuito. Un circuito con n variables de estado binarias tiene 2^n estados posibles. Como en la práctica 2^n es un valor finito, a los circuitos lógicos secuenciales también se les conoce como máquinas de estado finito, o simplemente *máquinas de estado*. [1]

Verilog es un lenguaje de descripción de hardware introducido por Gateway Design Automation en 1984. En 1995, la IEEE publicó el estándar IEEE 1364-1995, estandarizándose así por primera vez el lenguaje. En 2009, se publicó el estándar IEEE 1800-2009, mejor conocido como SystemVerilog. Actualmente, existe el estándar IEEE 1800-2023, publicado en el mes de diciembre del 2023. Verilog permite diseñar, simular y sintetizar circuitos digitales, incluido máquinas de estado. [1]

II. Objetivos

Objetivo general:
Diseñar el sistema de seguridad de la empresa Patitos S.A.

Objetivos específicos:
Implementar una máquina de estado en Verilog que satisfaga las condiciones de diseño descritas por la empresa.
Simular el sistema de seguridad en un entorno virtual para verificar su correcto funcionamiento.
Diseñar un teclado para el ingreso de la contraseña.

III. Metodología

Se utilizó Visual Studio Code para crear el código en Verilog. Además, se utilizó gtkwave para simular la máquina de estado diseñada.

Se sometió el diseño a diez pruebas para verificar su correcto funcionamiento:

- *Prueba 1*. El sistema comienza desarmado y se arma correctamente.
- *Prueba 2*. El sistema comienza desarmado y hay una ventana abierta.
- *Prueba 3*. El sistema comienza armado y se desarma correctamente.
- *Prueba 4*. El sistema comienza armado y se abre una ventana.
- *Prueba 5*. El sistema comienza armado y se abre una puerta, el usuario ingresa la contraseña correcta.
- *Prueba 6*. El sistema comienza armado y se abre una puerta, el usuario ingresa la contraseña incorrecta dos veces.
- *Prueba 7*. El sistema comienza armado y se activa la alarma contra incendios.
- *Prueba 8*. El sistema comienza desarmado y se activa la alarma contra incendios.
- *Prueba 9*. El sistema comienza armado, se detecta movimiento y se ingresa la contraseña correcta.
- *Prueba 10*. El sistema comienza en estado de incendio y se ingresa la contraseña correcta.

Se diseñó un teclado que contiene los diez números decimales y un Enter.
Se sometió al teclado a tres pruebas para verificar su correcto funcionamiento:

- *Prueba 11.* Se ingresa la contraseña correcta.
- *Prueba 12*. Se ingresa la contraseña incorrecta una vez, y luego se ingresa la contraseña correcta.
- *Prueba 13.* Se ingresa la contraseña incorrecta dos veces.

IV. Resultados y Discusión

Prueba 1.

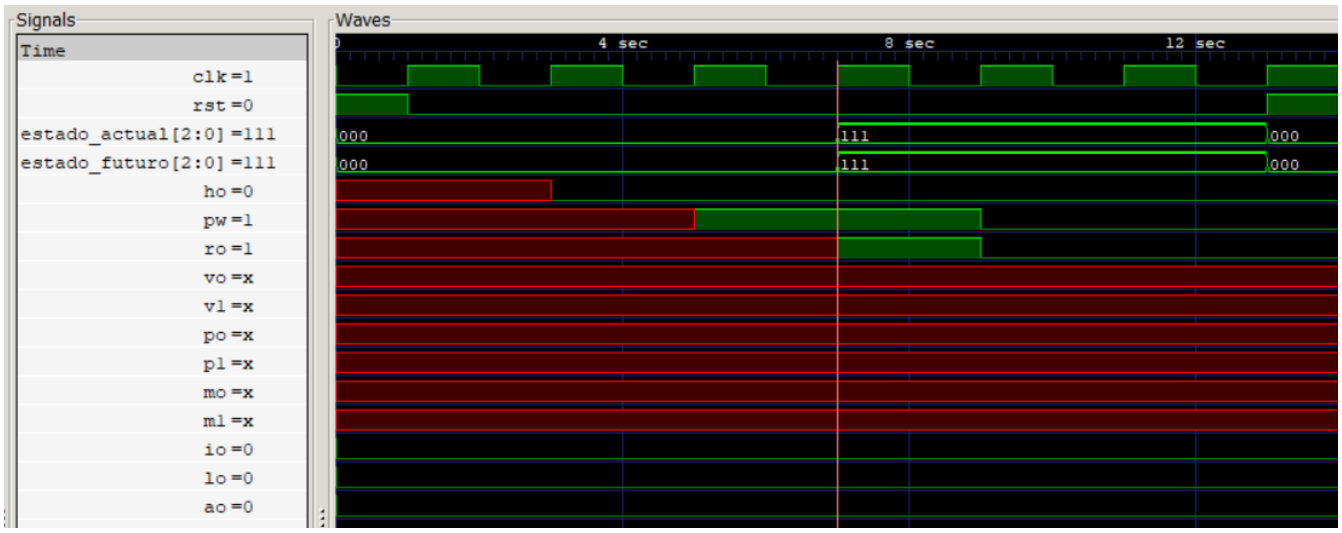

Figura 1. Resultados de la Prueba 1.

El sistema comienza en el estado A=000 (estado desarmado). La entrada de la señal de humo (h0) es 0, la contraseña es correcta (pw=1) y ha terminado el tiempo del cronómetro (r0=1). En consecuencia, se pasa al estado B=111 (estado armado). Finalmente, se desactivan las entradas para que no interfieran en las demás pruebas. Ninguna salida (I0, L0, A0) se ha activado aún.

Prueba 2.

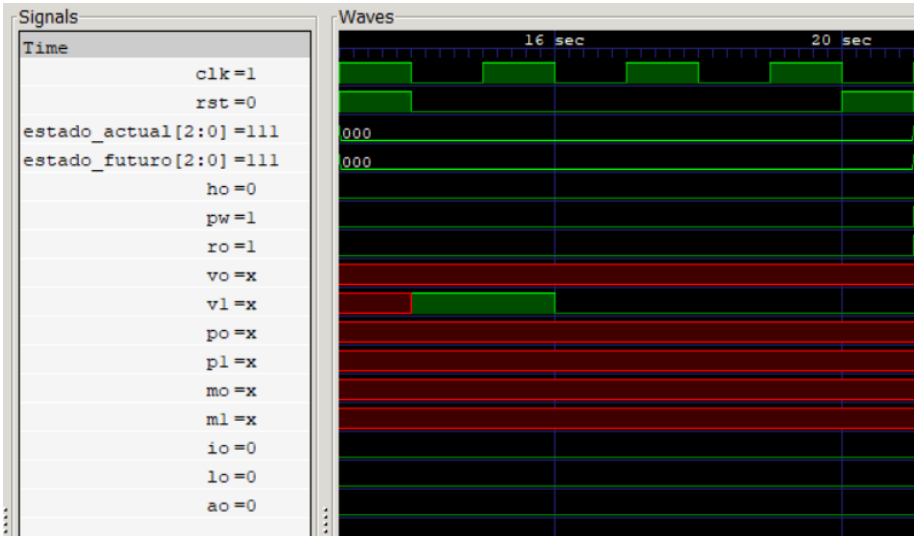

Figura 2. Resultados de la Prueba 2.

Se activa el reset para que el sistema de seguridad vuelva a su estado inicial (estado desarmado). Seguidamente, se abre la ventana v0. No se activó ninguna salida porque el sistema está desarmado. Finalmente, se cierra la ventana v0 para que no interfiera en las demás pruebas.

Prueba 3.

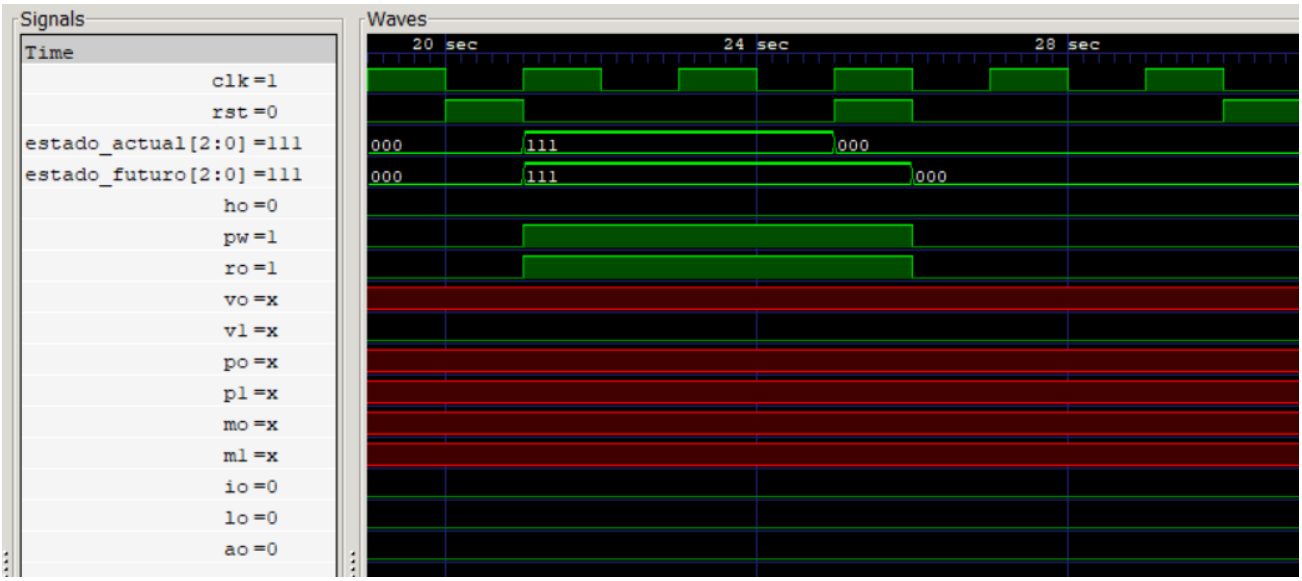

Figura 3. Resultados de la Prueba 3.

Se inicia en el estado B=111 (estado armado) y se activa el reset para regresar al estado A=000 (estado desarmado). Dado que no hay caminos para pasar directamente del estado a al b, se hace un reset.

Prueba 4.

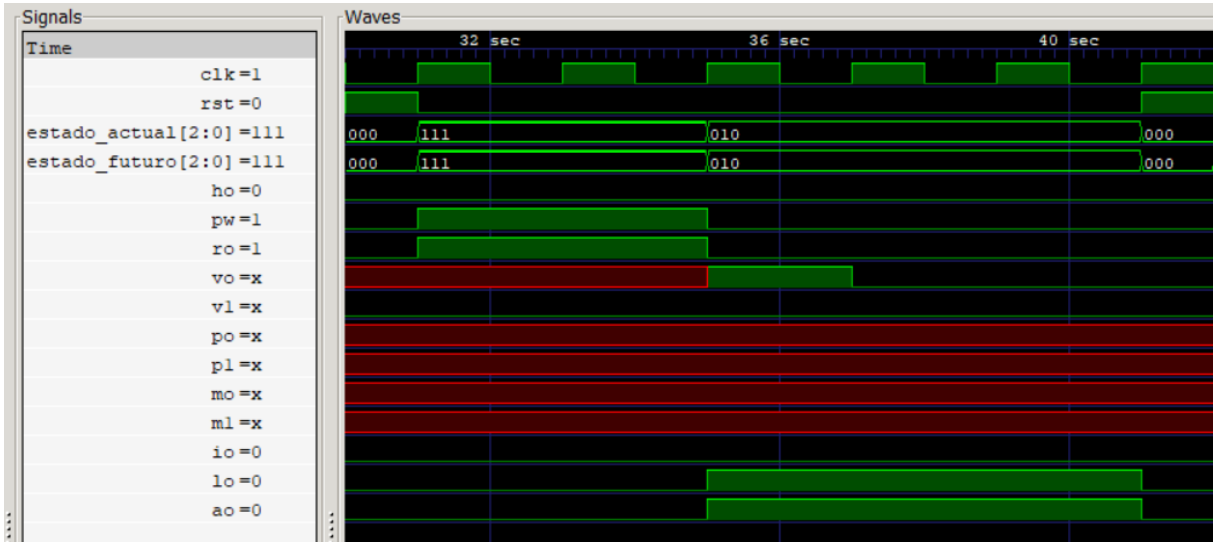

Figura 4. Resultados de la Prueba 4.

Se comienza en el estado B=111 (estado armado) y se abre una ventana (v0=1). El sistema de seguridad pasa al estado C=010 (estado de robo), el cual hace que se active la alarma de robo (A0) y las luces (L0).

Se desactiva la contraseña para comprobar que las luces y la alarma funcionan bien y finalmente se apagan las señales para que no influya en las pruebas restantes.

Prueba 5.

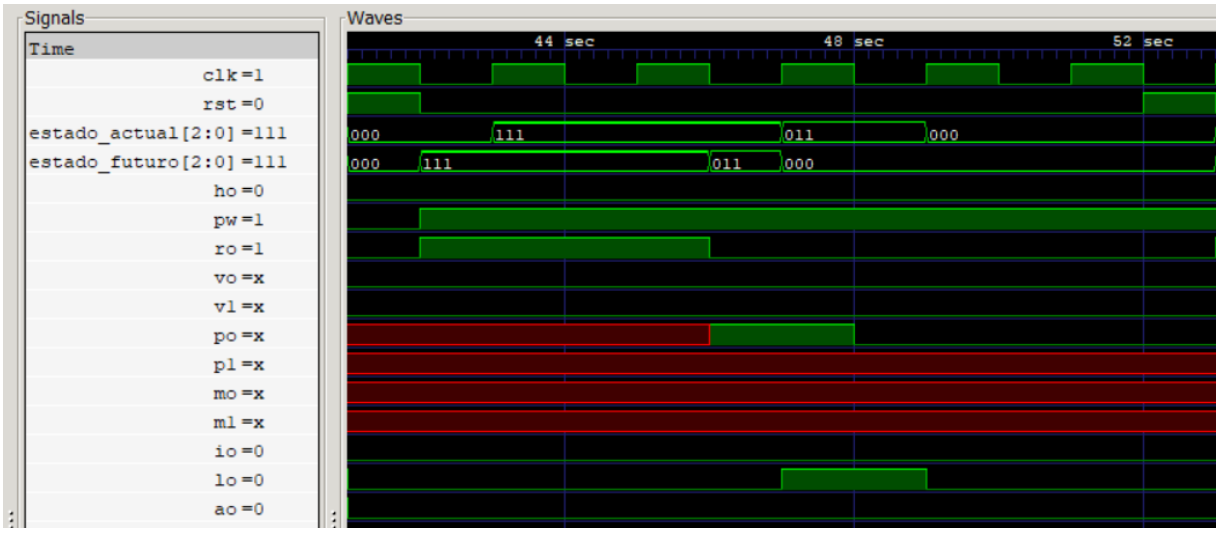

Figura 5. Resultados de la Prueba 5.

Se comienza en el estado B=111 (estado armado) y se abre una puerta (p0=1), el sistema pasa al estado D=011 (entrada autorizada) y se activan las luces, pero no la alarma. Enseguida, el usuario ingresa la contraseña correcta, las luces se apaguen (L0 desactivada) y el sistema vuelva al estado A=000 (estado desarmado). Durante el diseño, se toma la decisión de mantener encedida la entrada pw porque la intención de esta prueba es verificar que con la contraseña correcta se pase al estado desarmado.

Al final de esta prueba se apagan las señales r0 para que no detecte el estado armado y p0 para que no interfiera en los estados posteriores.

Prueba 6.

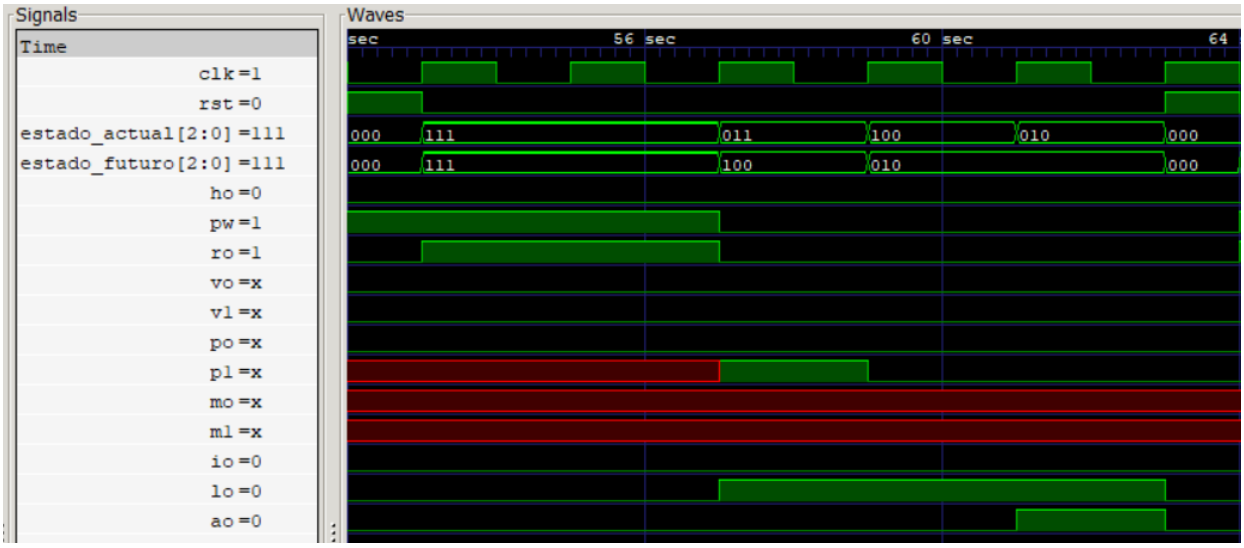

Figura 6. Resultados de la Prueba 6.

Se comienza en el estado B=111 (estado armado) y se abre una puerta (p1=1), el sistema pasa al estado D=011 (entrada autorizada) y se encienden las luces (L0), el usuario ingresa la contraseña incorrecta y el sistema pasa al estado E=100 para dar una segunda oportunidad de ingresar la contraseña correcta; nuevamente se ingresa la contraseña incorrecta y el sistema pasa al estado C=010 (robo), donde se activa la alarma de robo (A0).

Al igual que en los estados anteriores se apagan las señales para que no interfieran en las siguientes pruebas.

Prueba 7.

Figura 7. Resultados de la Prueba 7.

Se comienza en el estado B=111 (estado armado) y se detecta una señal de humo (h0=1) lo cual hace que el sistema pase al estado F=001 (incendio) y se active la alarma contra incendios (I0) y las luces (L0).

Para finalizar dicha prueba y aprobar su funcionamiento se desactivan las señales.

Prueba 8.

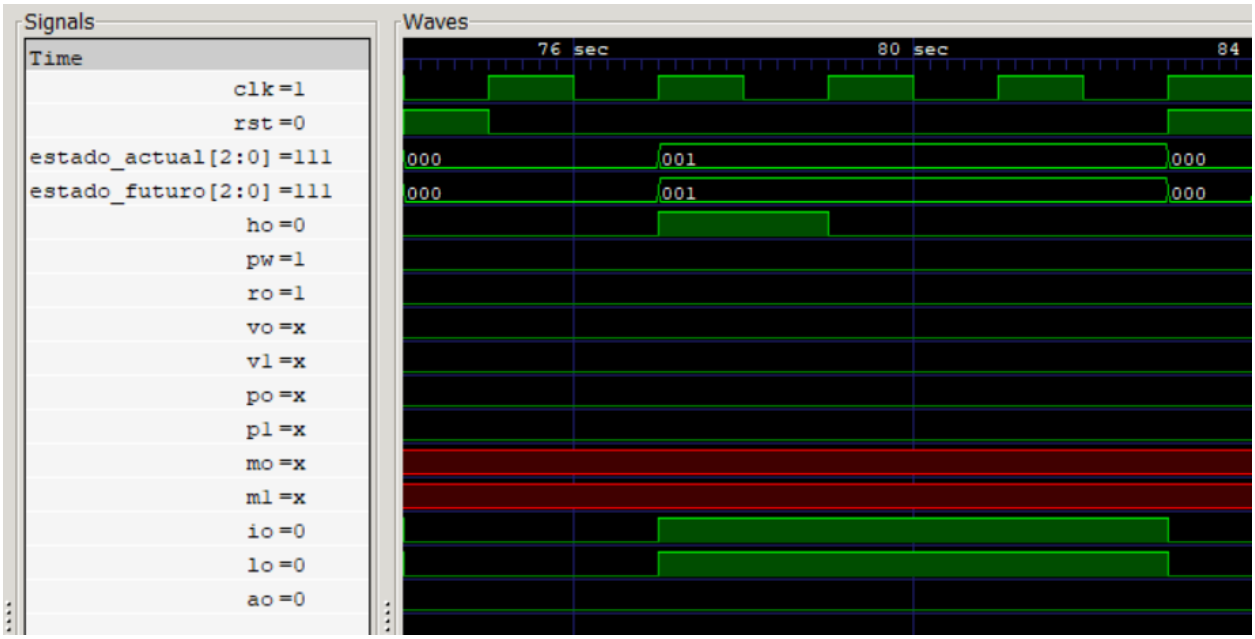

Figura 8. Resultados de la Prueba 8.

Se comienza en el estado A=000 (estado desarmado) y el sistema detecta una señal del humo que hace que la entrada h0 se active y se pase al estado F=001, encendiendo la alarma contra incendios I0 y las luces L0. Esta prueba adicional se realiza porque se requiere comprobar que la salida I0 se encienda ante la señal de humo aunque el sistema esté en estado desarmado.

Al terminar se apagan las señales para continuar con las pruebas.

Prueba 9.

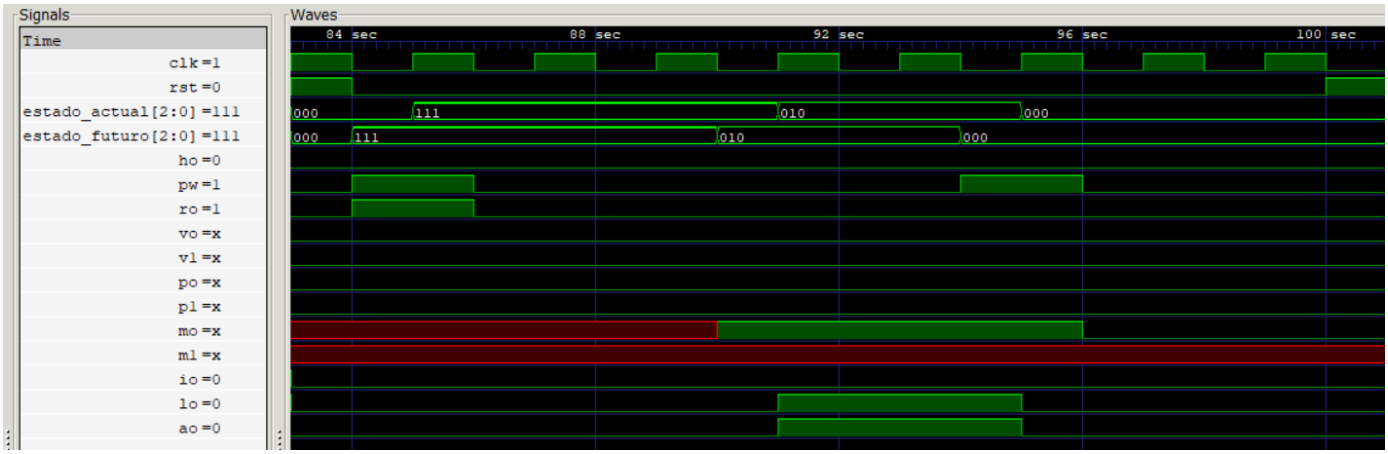

Figura 9. Resultados de la Prueba 9.

El sistema comienza en el estado B=111 (estado armado) y uno de los sensores detecta movimiento; la entrada M0 recibe un uno, lo que hace que el sistema pase al estado C=010 y se active la alarma de robos (A0) y las luces (L0). Esta prueba adicional se realiza para comprobar la correcta funcionalidad del sensor de movimiento.

Al finalizar la prueba se aprueba el funcionamiento del sensor y se apagan las señales.

Prueba 10.

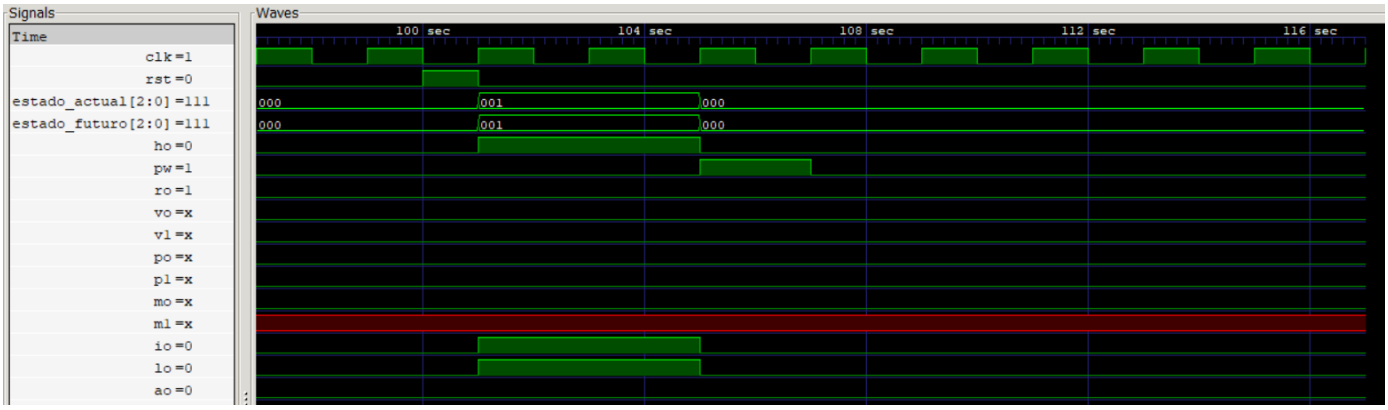

Figura 10. Resultados de la Prueba 10.

Para la prueba final el estado comienza detectando la señal de humo y hace que la entrada h0 sea igual a 1 y el sistema pase al estado F=001 (incendio) con la alarma de incendio (I0) activa, para desactivarla se debe ingresar la contraseña correcta (pw=1). Esta prueba adicional se realiza para comprobar que se puede devolver el sistema del estado de incendio al estado desarmado, en caso de que se detecte una falsa alarma de humo.

Prueba 11.

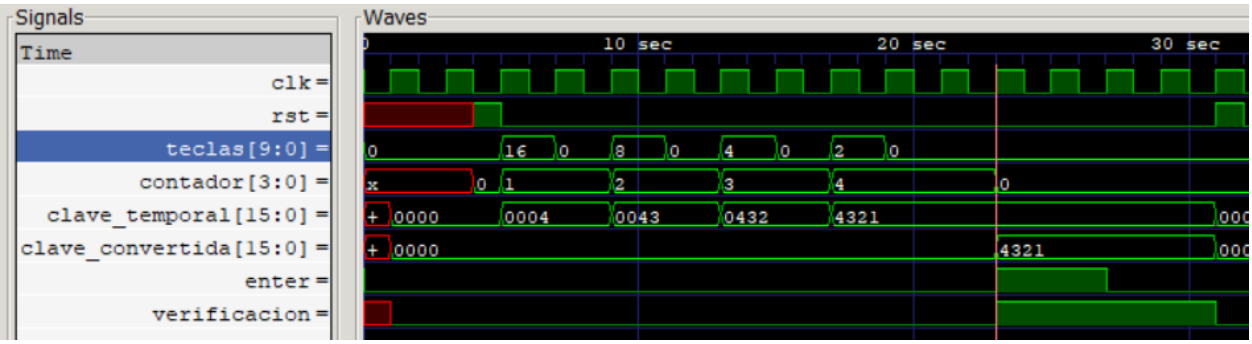

Figura 11. Resultados de la Prueba 11.

Se ingresa la contraseña correcta (4321). Nótese que las entradas están en BCD (2^4=16, 2^3=8, 2^2=4, 2^1=2). Se verifica el acceso.

Prueba 12.

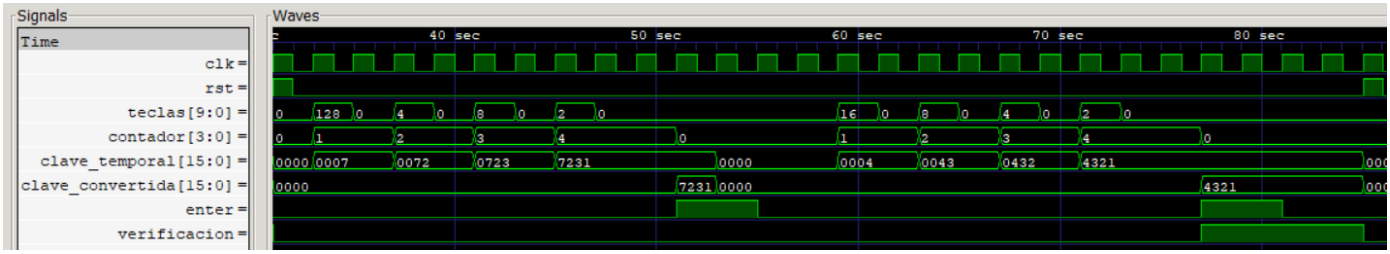

Figura 12. Resultados de la Prueba 12.

Se ingresa la contraseña incorrecta (7231). Nótese que las entradas están en BCD (2^7=128, 2^2=4, 2^3=8, 2^1=2). Posteriormente, se ingresa la contraseña correcta (4321). Se verifica el acceso.

Prueba 13.

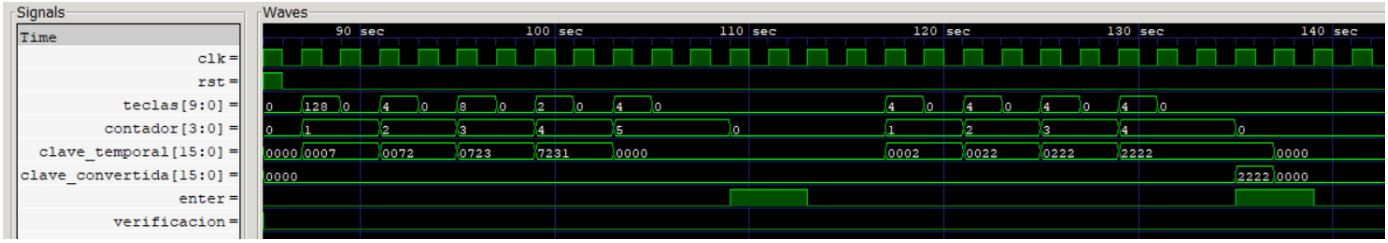

Figura 13. Resultados de la Prueba 13.

Se ingresa la contraseña incorrecta dos veces (7231 y 2222). No se verifica el acceso.

VI. Conclusiones

Verilog es un lenguaje de descripción de hardware que permite traducir problemas de la vida real en un modelo digital mediante el uso de máquinas de estado, y en consecuencia permite simular diferentes escenarios para diseñar posibles soluciones a los problemas.

Se logró diseñar y simular el sistema de seguridad y el teclado de contraseña para la empresa Patitos S.A. y pasó las pruebas necesarias para su implementación.

Referencias

[1] J. F. Wakerly, “Digital Design: Principles and Practice”. Quinta edición. Nueva Jersey, Estados Unidos: Pearson Education, 2018.