

Tarea #1

Descripción conductual de un controlador automatizado para la entrada de un estacionamiento

Especificaciones

1. Su objetivo es diseñar un controlador automatizado para la entrada de un estacionamiento de acuerdo con las especificaciones de este enunciado:
 - a. A la entrada del estacionamiento hay un sensor que se activa para detectar la llegada de un vehículo.
 - b. Una vez que se activa el sensor del inciso “a”, se le solicita a quien conduce el vehículo una clave de cuatro dígitos, representados como 16 bits en BCD (cuatro bits por dígito).
 - c. La clave digitada llega al controlador como una sola señal de 16 bits en paralelo.
 - d. La clave correcta debe ser un parámetro de entrada para su módulo de Verilog, y debe definirse como tal en la instancia del testbench. La clave por defecto del módulo debe ser 2468.
 - e. Si la clave proporcionada es correcta, el controlador debe producir una señal para abrir la compuerta para dejar entrar al vehículo.
 - f. Si la clave proporcionada es incorrecta, la compuerta debe permanecer cerrada.
 - g. Si se ingresa una clave incorrecta tres veces seguidas, se debe activar una alarma de pin incorrecto y el sistema debe quedar bloqueado hasta que se reinicie.
 - h. Cuando se ingresa la clave correcta y no se ha alcanzado el máximo de intentos, se debe limpiar el contador de intentos incorrectos.
 - i. Existe otro sensor para indicar que el auto ya terminó de entrar y que se debe cerrar la compuerta.
 - j. Al activarse el sensor del inciso “i” se debe desactivar la señal para abrir la compuerta y activar otra señal para cerrar la compuerta.
 - k. Si en algún momento se encienden al mismo tiempo el sensor del inciso “a” y el sensor del inciso “i”, la compuerta debe bloquearse y encender una alarma de bloqueo.
 - l. Si ocurren las condiciones del inciso “k”, se debe permanecer en el estado de bloqueo hasta que se reinicie el sistema.
 - m. Para efectos de las pruebas, la clave correcta debe corresponder a los últimos cuatro dígitos de su número de carné representados como un número binario de 16 bits en BCD.

Trabajo a realizar sobre el dispositivo a diseñar

Su diseño debe incluir al menos los siguientes entregables:

- Diagrama de bloques detallando entradas y salidas del sistema.
- Diagrama de estados o diagrama ASM para representar el comportamiento del controlador
- Plan de pruebas mínimo para verificar el comportamiento correcto del controlador
- Código de Verilog conductual para implementar el módulo del controlador.
- Código del banco de pruebas (testbench) necesario para implementar el plan de pruebas propuesto.
- Resultados de las simulaciones conductuales que ejecutan el plan de pruebas propuesto (utilizando Icarus Verilog o algún simulador similar).

Propuesta de Plan de Pruebas Mínimo

El conjunto de pruebas mínimas que se deben hacer para validar el diseño conductual solicitado por lo menos debe incluir:

1. **Prueba #1, funcionamiento normal básico.** Llegada de un vehículo, ingreso del pin correcto y apertura de puerta, sensor de fin de entrada y cierre de compuerta.
2. **Prueba #2, ingreso de pin incorrecto menos de 3 veces.** Llegada de un vehículo, ingreso de pin incorrecto (una o dos veces), puerta permanece cerrada. Ingreso de pin correcto, funcionamiento normal básico. Revisión de contador de intentos incorrectos.
3. **Prueba #3, ingreso de pin incorrecto 3 o más veces.** Revisión de alarma de pin incorrecto. Revisión de contador de intentos incorrectos. Ingreso de pin correcto, funcionamiento normal básico. Revisión de limpieza de contadores y alarmas.
4. **Prueba #4, alarma de bloqueo.** Ambos sensores encienden al mismo tiempo, encendido de alarma de bloqueo, ingreso de clave incorrecta, bloqueo permanece. Ingreso de clave correcta, desbloqueo. Funcionamiento normal básico.

Entregables:

Con el fin de facilitar el proceso de revisión, se le solicita entregar el proyecto en EDA playground o bien organizar los entregables de la tarea de la siguiente manera:

- a) Entregar un solo archivo comprimido, y nombrado según el patrón <# de carné>.<formato de compresión>, por ejemplo C21047.zip
- b) El archivo comprimido descrito en el rubro a) deberá contener específicamente los siguientes archivos:
 - Reporte en formato PDF cuyo nombre debe seguir el patrón <# de carné>.pdf, por ejemplo, C21047.pdf
 - Uno o varios archivos de Verilog con el formato <nombre de archivo>.v que construyan la solución que se solicita en la tarea.
 - Un solo archivo probador llamado tester.v
 - Un solo archivo de banco de pruebas llamado testbench.v
 - Un archivo Makefile que permita correr todos los pasos de simulación con una sola línea de comando.

- c) El banco de pruebas, testbench.v, deberá incluir a todos los demás archivos *.v, de modo que la compilación y simulación del testbench implique la compilación y simulación de todos los archivos internos.
- d) El probador, tester.v, debe escribirse de forma tal que una sola simulación contenga los resultados de todas las operaciones que ejemplifiquen el funcionamiento esperado del módulo.
- e) El Makefile debe contener, como mínimo, los comandos necesarios para correr la compilación y simulación de los módulos de la tarea. La figura #1 muestra un ejemplo general como guía. Note que este ejemplo incluye también la síntesis, que NO se solicita en esta tarea.

```
tarea: testbench.v mdio.js #Archivos requeridos
      yosys -s mdio.js      #Corre síntesis
      iverilog -o salida testbench.v #Corre Icarus
      vvp salida #Corre la simulación
      gtkwave resultados.vcd #Abre las formas de onda
```

Figura #1: Ejemplo de Makefile

Rúbrica de Calificación

Tarea #1: Descripción conductual de un control de estacionamiento	Categoría	% Categoría	% Rubro	% Total
Existe una descripción conductual en Verilog del controlador de estacionamiento. Esta descripción incluye al menos un módulo de banco de pruebas (testbench.v), un módulo probador (tester.v) y un módulo para el dispositivo bajo prueba (DUT).	Código	20%	20%	4%
Se construye un banco de pruebas para este controlador y se incluye los módulos adicionales necesarios para probar su correcto funcionamiento.	Código	20%	20%	4%
Las descripciones de Verilog se entregan en archivos distintos al reporte, listos para ser simulados, e incluyen un archivo de Makefile, de modo que la simulación se corre con una sola línea de comando. Alternativamente, se entregan como un proyecto en EDA playground	Código	20%	10%	2%
Las descripciones en Verilog están comentadas adecuadamente para que otras personas entiendan la lógica de la descripción.	Código	20%	10%	2%
Las descripciones en Verilog compilan sin producir errores.	Código	20%	20%	4%
Las descripciones en Verilog ejecutan correctamente. Es decir, corren, entregan algunos resultados y finalizan.	Código	20%	20%	4%
El controlador completa el funcionamiento normal básico de forma correcta de acuerdo con la especificación dada.	Pruebas	60%	50%	30%
El controlador reacciona de forma correcta al ingreso de pines incorrectos de acuerdo con la especificación dada.	Pruebas	60%	25%	15%
El controlador reacciona ante la situación de bloqueo de forma correcta de acuerdo con la especificación dada.	Pruebas	60%	25%	15%
El reporte contiene las siguientes secciones: Resumen, descripción arquitectónica, plan de pruebas, instrucciones de utilización de la simulación para quien califica, ejemplos de resultados, conclusiones y recomendaciones.	Reporte	20%	40%	8%
El reporte explica con claridad los detalles relevantes del diseño particular que se hizo, las partes del diseño que dieron más trabajo para completar y por qué, una explicación de los problemas que se presentaron y cómo se solucionaron.	Reporte	20%	40%	8%
La longitud del reporte no excede 10 páginas.	Reporte	20%	20%	4%

Guía para el reporte

Se debe entregar en forma electrónica un documento, a lo sumo de 10 páginas de longitud, que incluya los siguientes puntos:

1. **Resumen:** Breve (Media página máximo) descripción de todo el proyecto. Esta sección es fundamental pues puede determinar si el lector se interesa o no en leer los detalles del proyecto. Un resumen mal hecho puede esconder un excelente proyecto. El resumen debería incluir:
 - a) Descripción breve del sistema, es decir, qué hace. Incluya alguna característica que considere que distingue este diseño en particular.
 - b) Las pruebas que se realizaron y qué resultados se obtuvieron. Indique problemas que se tuvieron que considere importante resaltar.
 - c) Conclusiones más importantes y recomendaciones para un diseño posterior.
2. **Descripción Arquitectónica:** Incluye un diagrama de bloques con las señales más importantes que sirve como base para describir el funcionamiento del sistema. La descripción va en términos de lo que se espera que el sistema haga. Es decir, se debe detallar la funcionalidad del sistema, el protocolo de las señales que se usan para que funcionen cada una de las partes y las secuencias de eventos que se deben dar. Esta descripción podría ir acompañada de tablas de verdad, tablas de transición de estados, diagramas de estados, diagramas temporales, etc.
3. **Plan de Pruebas:** Aquí se deben enumerar, esto es, se debe presentar una **lista detallada** de las pruebas que se le van a hacer al diseño para verificar que está funcionando de acuerdo a las especificaciones dadas. La lista debe contener por lo menos los siguientes elementos i) Nombre/número de prueba, ii) Descripción de la prueba, y iii) Una indicación de si el diseño la falló o la pasó. Estas pruebas podrían incluir la generación de vectores de entrada para probar en forma exhaustiva todas las líneas de una tabla de verdad o tabla de estados, patrones aleatorios de entradas para tratar de causar errores en la respuesta del diseño, o patrones específicos que ejerciten un cierto modo de funcionamiento. Cada prueba debería ser claramente enumerada en el plan para que también se pueda hacer referencia a ella en el código del banco de pruebas del diseño.
4. **Instrucciones de utilización de la simulación:** Esta sección debe mostrar los comandos necesarios para hacer funcionar la simulación en todos los casos que especifica el plan de pruebas. Hay que suponer que el diseño de un grupo puede ser utilizado por otro grupo o el profesor. Si los resultados no se pueden repetir porque no se conocen los comandos para hacer funcionar la simulación entonces es como si el diseño no funcionara del todo. Se recomienda crear un Makefile de modo que se pueda correr todas las pruebas del caso con un solo comando en Icarus Verilog y GTKwave.
5. **Ejemplos de los resultados:** Una descripción de los resultados más importantes acompañados de los diagramas temporales de la simulación (GTKWave) o cualquier otra salida que demuestre claramente el comportamiento descrito. No es necesario incluir una muestra exhaustiva de resultados, sino que los más representativos del diseño. El punto es mostrarle al lector los comportamientos más sobresalientes para formarle una idea clara

del funcionamiento del diseño. Ya verá el lector si desea más detalles, entonces podrá correr una simulación.

6. **Conclusiones y recomendaciones:** Basado en los resultados obtenidos se indica aquí qué se logró con el proyecto. Puede ser que se concluya que con el diseño propuesto se tiene una limitación en la velocidad de respuesta de... etc. O que con ciertas combinaciones de entradas el diseño se vuelve inestable o los resultados no son los esperados. También se puede concluir qué ventajas o problemas encontraron al seguir el plan de trabajo. A raíz de las conclusiones se puede también recomendar cómo se podría mejorar el diseño o qué otras pruebas se le podrían hacer para garantizar su funcionamiento en otras condiciones que al principio no se consideraron, o también cómo se debería planear el siguiente proyecto para poder cumplirlo a tiempo.