

Tarea #1

Descripción conductual de un controlador automatizado para una máquina expendedora de café

Especificaciones

1. Su objetivo es diseñar un controlador automatizado para una máquina expendedora de café de acuerdo con las especificaciones de este enunciado:
 - a. La máquina expendedora de café puede entregar distintos tipos de café según se muestra en la tabla #1.

Tipo de café	Tamaño	Concentración	Leche?	Espuma?	Nivel de azúcar	Precio
Negro	Pequeño	Regular	No	No	De 0 a 5	500 colones
Negro	Mediano	Regular	No	No	De 0 a 5	1000 colones
Negro	Grande	Regular	No	No	De 0 a 5	1500 colones
Con leche	Pequeño	Regular	Sí	No	De 0 a 5	750 colones
Con leche	Mediano	Regular	Sí	No	De 0 a 5	1250 colones
Con leche	Grande	Regular	Sí	No	De 0 a 5	1750 colones
Espresso	Pequeño	Fuerte	No	No	De 0 a 5	1000 colones
Espresso	Mediano	Fuerte	No	No	De 0 a 5	1500 colones
Espresso	Grande	Fuerte	No	No	De 0 a 5	2000 colones
Capuccino	Pequeño	Fuerte	Sí	Sí	De 0 a 5	1250 colones
Capuccino	Mediano	Fuerte	Sí	Sí	De 0 a 5	1750 colones
Capuccino	Grande	Fuerte	Sí	Sí	De 0 a 5	2250 colones

Tabla #1: Opciones de café disponibles

- b. Para comprar un café, el usuario selecciona, a través de una pantalla, el tipo de café, el tamaño y el nivel de azúcar. El controlador que usted debe diseñar recibe esas tres entradas como códigos binarios que usted debe asignar según su diseño. NO se le solicita el diseño de la pantalla, puede suponer que recibe las entradas como códigos binarios directamente.
 - c. Una vez seleccionadas las tres opciones de entrada, se debe desplegar el precio de la opción seleccionada en la pantalla. Para esto, el controlador debe producir una salida binaria que represente el monto en colones correspondiente al precio. De nuevo, NO se le solicita el diseño de la pantalla, el controlador de su diseño solo debe producir una salida correspondiente al monto, codificada en binario.
 - d. El controlador debe permanecer en el estado de despliegue del precio hasta que reciba el pago correspondiente o hasta que expire un temporizador de espera. Para indicar la recepción del pago, se cuenta con una entrada binaria de PAGO_RECIBIDO.

- e. El temporizador de espera es un contador que se debe incrementar en una unidad por cada ciclo de reloj que pase desde el momento en que se espera el pago hasta que se recibe la indicación de PAGO_RECIBIDO.
- f. El tiempo de expiración del temporizador corresponde a los últimos dos dígitos de su número de carné. Por ejemplo, si su número de carné fuera C11047, el valor necesario para que el temporizador expire sería 47. Si el penúltimo dígito de su número de carné es cero, debe sustituirlo por 9, de modo que la combinación 00, por ejemplo, se convertiría en 90.
- g. En caso de que el temporizador expire, se debe regresar al estado inicial de selección de la bebida.
- h. En caso de que se reciba un pago antes de que expire el temporizador, se debe pasar a un estado de preparación de café, donde se produzcan salidas binarias correspondientes a la concentración, leche, espuma y nivel de azúcar, según las opciones seleccionadas por el usuario.
- i. El tiempo que debe sostenerse el valor de las salidas del inciso h está determinado por el tamaño de la bebida seleccionada. Pequeño, las salidas se sostienen por un ciclo de reloj. Mediano, se sostienen por dos ciclos de reloj. Grande, se sostienen por tres ciclos de reloj.
- j. Una vez preparada la bebida se debe regresar al estado inicial.

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.** Se debe verificar que para todos los tipos de café que se pueden seleccionar se despliega el precio correcto y que, si se recibe el pago normalmente, se producen las salidas correctas para la preparación del café.
2. **Prueba #2, expiración del temporizador.** Se debe verificar que si el temporizador expira se regresa al estado inicial y se puede iniciar una nueva transacción sin que las selecciones anteriores interfieran con el nuevo proceso.
3. **Prueba #3, dos transacciones correctas seguidas.** Se debe verificar que al completar una transacción válida correctamente, el sistema regresa al estado inicial y se puede realizar una segunda transacción válida correctamente sin interferencia de la primera.
4. **Pruebas específicas del diseño.** Como cada diseño es diferente, es recomendable añadir pruebas adicionales particulares, que verifiquen aspectos específicos de cada diseño individual.

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 para una máquina expendedora de café	Categoría	% Categoría	% Rubro	% Total
Existe una descripción conductual en Verilog para el controlador de la máquina de café. 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	10%	20%	2%
Se define un plan de pruebas para este controlador y este incluye al menos el plan de pruebas mínimo recomendado, así como pruebas adicionales específicas del diseño.	Código	10%	20%	2%
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	10%	10%	1%
Las descripciones en Verilog están comentadas adecuadamente para que otras personas entiendan la lógica de la descripción.	Código	10%	10%	1%
Las descripciones en Verilog compilan sin producir errores.	Código	10%	20%	2%
Las descripciones en Verilog ejecutan correctamente. Es decir, corren, entregan algunos resultados y finalizan.	Código	10%	20%	2%
El controlador completa el funcionamiento normal básico de forma correcta de acuerdo con la especificación dada.	Pruebas	70%	40%	28%
El controlador reacciona de forma correcta ante el escenario de expiración del temporizador	Pruebas	70%	20%	14%
El controlador reacciona de forma correcta al efectuar dos transacciones seguidas, de acuerdo con la especificación dada.	Pruebas	70%	20%	14%
Las pruebas específicas del diseño definidas en el plan de pruebas se ejecutan correctamente.	Pruebas	70%	20%	14%
El reporte contiene las siguientes secciones debidamente desarrolladas: Resumen, descripción arquitectónica, plan de pruebas, instrucciones de utilización de la simulación para quien califica, ejemplos de resultados, conclusiones y recomendaciones.	Reporte	20%	30%	6%
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%	60%	12%
La longitud del reporte no excede 10 páginas.	Reporte	20%	10%	2%

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.