
Laboratorio 1: Introducción a los lenguajes de descripción de hardware

Fecha de asignación: 28 febrero 2025
Grupos: 3 personas

Fecha de entrega: 12 marzo 2025
Profesor: Luis Barboza, Leonardo Araya

1. Introducción

Este laboratorio introduce al estudiante al diseño de circuitos digitales por medio de lenguajes de descripción de hardware (HDL). Un HDL es una forma de describir la especificación, comportamiento o estructura de un módulo de hardware, por medio de una estructura programática. Pese a su similitud con un lenguaje de programación (tipos de datos, estructuras básicas, sintaxis, etc) los HDL **deben describir hardware**, por lo que para todo diseño, se debe tener conocimiento completo de qué componente se está diseñando. Esto implica la realización previa de tablas de verdad, diagramas de estados, diagramas de bloques, etc.

En este laboratorio se trabajarán los lenguajes *System Verilog* y VHDL. Para más información sobre los lenguajes a utilizar, se recomienda estudiar el capítulo 4 de [?].

1.1. Procedimiento

A lo largo del laboratorio se desarrollarán varios ejercicios que guiarán a los estudiantes por el proceso de diseño, implementación y verificación de circuitos digitales básicos, por medio de lenguajes de descripción de hardware.

1. Lea y trate de comprender todo el trabajo solicitado antes de iniciarlo.
2. Para la presentación funcional, se le pedirá que muestre algunos de los ejercicios propuestos, ya sea en implementación o simulación. Dicha selección se hará el día de la presentación.

2. Investigación

Para el desarrollo de este laboratorio se deben responder las siguientes preguntas.

1. Explique el modelado de comportamiento y de estructura en diseño digital. Brinde un ejemplo de cada uno.
2. Explique el proceso de síntesis lógica en el diseño de circuitos digitales.

3. Investigue sobre la tecnología de FPGAs. Describa el funcionamiento de la lógica programable en general, así como los componentes básicos de una.
4. Investigue sobre los actuales modelos de FPGA utilizados en la industrias.
5. Investigue sobre cuáles son las aplicaciones más comunes en la industria que tienen las FPGA.

3. Ejercicios

A continuación se presentan 3 ejercicios prácticos, los cuales debe resolver de manera completa.

3.1. Problema 1

1. Diseñe un decodificador que permita convertir un número de 4 bits en binario a su respectiva representación en BCD (*binary-coded decimal*), mediante modelo de comportamiento en SystemVerilog. El diseño debe contemplar las tablas de verdad. **No se permite el uso de *case* para convertir el número a BCD.**
2. Realice un *testbench*, en SystemVerilog, en el que se muestre de manera simple el funcionamiento del decodificador. Muestre el resultado de la prueba para al menos 8 valores diferentes.
3. Implemente el decodificador en la FPGA. Utilice los switches (iniciando en el cero) como entrada y dos displays para 7 segmentos para visualizar el funcionamiento del decodificador. El resultado se mostrará en decimal.

3.2. Problema 2

1. Diseñe un sumador completo de 4 bits con modelo de estructura en VHDL. Parta del diseño un sumador completo de 1 bit.
2. Realice un *testbench* para el sumador del punto anterior, en VHDL. Muestre las pruebas para al menos 4 valores diferentes de operandos.
3. Implemente el sumador completo en FPGA. Utilice los switches de las tarjeta para los datos de entrada, así como los display de 7 segmentos para mostrar el resultado correcto (en hexadecimal).

3.3. Problema 3

1. Diseñe un restador parametrizable de N bits con reset asincrónico en el lenguaje de descripción de su preferencia.
2. Realice un *testbench* de auto-chequeo (self-checking) para el contador del punto anterior. Muestre el resultado del *testbench* para 2, 4 y 6 bits.
3. Implemente el contador para 6 bits en la FPGA, utilice un botón para decrementar, un switch como reset, así como los displays de 7 segmentos para visualizar el funcionamiento del restador (en decimal o hexadecimal). Además utilice los botones para establecer el valor inicial del restador regresivo.

4. Metodología de trabajo

El proyecto debe seguir los siguientes aspectos de desarrollo, sino, la parte funcional no será calificada y obtendrá nota de cero:

1. Utilice una cuenta de repositorio gratuita.
2. Cree un repositorio por grupo con el siguiente nombre:
`<user_id1>_<user_id2>_<user_id3>_digital_design_lab_2024`. El `user_id` estará compuesto por la primera letra del nombre y el apellido. Por ejemplo, para el estudiante Luis Barboza, el nombre del repositorio será:
`lbarboza_integrante2_integrante3_digital_design_lab_2024`.
3. Si el repositorio es privado, proporcione acceso a `abner2111` (GitHub y GitLab).
4. El repositorio de Git contendrá dos ramas principales: `master` y `development`.
5. Inicialmente, la rama de `development` se crea a partir del `master`.
6. Al trabajar en un proyecto, el estudiante debe crear una nueva rama de trabajo desde `develop` y cuando la función esté lista, la rama debe fusionarse para `develop`. Cualquier corrección o modificación adicional después de `merge` debería requerir que se repita el proceso (es decir, crear la rama desde `develop` y fusionar los cambios más tarde). Una vez que el código de desarrollo esté listo, se fusionará con `master` y se debe crear una `tag`. El proceso se describe en la siguiente Figura 1.

Adicionalmente se coloca este [enlace recomendado](#).

7. Después de haber realizado algunos laboratorio la rama `master` debe verse así:

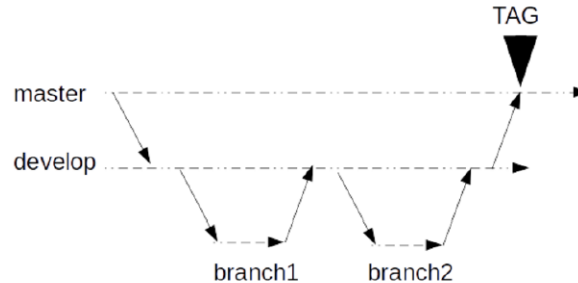


Figura 1: *Git workflow*

- master/
 - laboratorio_1
 - laboratorio_2
 - ...
- ...

Donde cada directorio de `laboratorio_x` contiene todos los entregables para cada laboratorio.

No es permitido realizar todo el trabajo en un solo commit, es decir, que realice el trabajo de forma local y solo suba el último entregable en el repositorio. Si no, obtendrá nota de cero. Debe mostrar avance incremental (se revisarán estadísticas).

5. Evaluación y entregables

5.1. Informe

Se deberá realizar un informe en formato IEEE en que se presente una sección de **introducción** al laboratorio con los conceptos teóricos y metodológicos relevantes (pueden utilizar la investigación); otra sección de **desarrollo** o presentación de cada ejercicio así como elementos de la solución y el diseño (**NO** incluir código fuente); una sección de **resultados** que describa los productos obtenidos del laboratorio (tablas, gráficos, etc), con su correspondiente **análisis** con respecto a la teoría (comparación teoría y práctica), y, finalmente, una sección de **conclusiones** en los que se determine cómo la solución planteada para los ejercicios hace que se cumplan los objetivos del laboratorio. Debe incluir además una sección de **referencias** en la que incluyan las fuentes a utilizar tanto en la sección de introducción como de análisis.

5.2. Presentación funcional

La presentación funcional se realizará en clase de manera sincrónica, para este deberá contar con los archivos necesarios de todos los ejercicios del laboratorio.

5.3. Bitácora

La bitácora será manejada por medio de Github. Se revisará un avance incremental en los commits y la participación de todos los integrantes. Debe subir el enlace en el apartado del tec digital.

5.4. Quiz individual

El día de la presentación funcional, antes de la misma, se asignará un quiz individual en el que se evaluarán algunos de los temas abarcados en este laboratorio, así como los resultados obtenidos del mismo.

Si tienen dudas puede escribir al profesor al [correo electrónico](#). **Los documentos serán sometidos a control de plagios.** La entrega se debe realizar por medio del TEC-Digital en la pestaña de evaluación. No se aceptan entregas extemporáneas después de la fecha de entrega a las 11:59 pm como máximo.