

## Proyecto II: Introducción a diseño digital en HDL

Prof. Kaleb Alfaro Badilla  
Dr.-Ing Alfonso Chacón-Rodríguez

### 1. Introducción

La demanda del mercado actual en la microelectrónica exige diseños digitales altamente complejos, que para implementarlos se requiere de ayuda asistida por computadora. Las herramientas EDA aumentan la productividad de los ingenieros al abstraer detalles de la implementación de los sistemas digitales por medio de un lenguaje de descripción de hardware (HDL) más natural para describir funcionalidades y comportamientos.

Los lenguajes HDL se utilizan para la síntesis de diseños digitales para ser fabricados en silicio o en FPGA. Las FPGAs, son circuitos integrados reconfigurables que de acuerdo su programación implementan una especificación funcional a partir de un código HDL.

### 2. Objetivo general

Introducir al estudiante al desarrollo de un sistema digital utilizando lenguajes de descripción de hardware.

### 3. Objetivos específicos

1. Elaborar una implementación de un diseño digital en una FPGA.
2. Construir un *testbench* básico para validar las especificaciones del diseño.
3. Implementar el algoritmo de generación de los códigos de Gray.
4. Coordinación de trabajo en equipo mediante el uso de herramientas de control de versiones.
5. Practicar planificación de tareas para trabajo de grupo.

### 4. Especificación del diseño

Se solicita el desarrollo de un circuito decodificador de Gray. El mismo deberá construirse según las pautas fundamentales de diseño digital síncrono. El circuito constará de tres subsistemas:

1. Subsistema de lectura y decodificación de código Gray.
2. Subsistema de despliegue de código ingresado traducido a formato binario en luces LED.
3. Subsistema de despliegue de código ingresado y decodificado en display de 7 segmentos.

Tabla 1: Código Gray de 4 bits a implementar.

Código decimal	Código binario de 4 bits	Código Gray de 4 bits
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

#### 4.1. Subsistema de lectura y decodificación de código Gray

El subsistema de lectura tomará de cuatro conmutadores el código ya generado según la tabla 1.

La entrada del código deberá ser capturada y sincronizada con el sistema principal. El muestreo de los mismos deberá realizarle al menos cada 500 ms.

El subsistema luego deberá traducir la entrada capturada a un código binario antes de enviarse al siguiente subsistema.

#### 4.2. Subsistema de despliegue de código ingresado traducido a formato binario en luces LED

Este subsistema toma los datos ya pasados a código binario y los despliega en cuatro luces LED. Este sistema deberá refrescar las luces al menos cada 500 ms.

#### 4.3. Subsistema de despliegue de código decodificado en display de 7 segmentos.

Este subsistema toma los datos en código binario y los despliega los dispositivos 7 segmentos disponibles en la placa, de forma decimal. El sistema deberá tener la tasa de refresco adecuada para una visualización cómoda por parte del usuario.

### 5. Guía de diseño

Los estudiantes deberán generar un diagrama de bloques para cada subsistema, con su funcionalidad descrita y su esquema de interconexión. Deberá presentarse adecuadamente la ruta de datos desde la salida hasta la entrada, con una descripción comportamental de cada sub-bloque dentro de los subsistemas (i.e., muxes, decos, registros, etc.). No será necesario llevar la descripción a álgebra booleana.

Cada subsistema deberá estar adecuadamente registrado a la entrada y salida. El flujo de datos debe ser indicado de manera explícita. No es necesario mostrar en detalle los bloques en que se implementen las máquinas de estados finitos (FSM) que se diseñen, pero sí sus diagramas de estado y las señales de control que manejan cada bloque en la ruta de datos.

### 5.1. Sobre la codificación del HDL

Se utilizarán SystemVerilog y el suite de herramientas de Vivado para desarrollar el sistema completo. Se seguirá el formato visto en clase para el estilo de codificación, cuyo apego por parte de los estudiantes formará parte de la evaluación final.

### 5.2. Sobre la verificación

Para cada subsistema, será obligatorio presentar simulaciones tanto a nivel de RTL (pre-síntesis) como con información de temporizado (post-síntesis y post-colocación-y-ruteo, las que serán evaluadas por el profesor de manera presencial en las fechas que se indicarán en el cronograma de la sección 6. Para ello, el grupo deberá generar las pruebas de banco o *testbenches* necesarios

Los grupos compartirán las FPGA para demostrar el funcionamiento de sus circuitos. Para ello el profesor armará una lista para distribuir las mismas. No se entregarán las placas hasta que al menos los grupos muestren simulaciones post-síntesis funcionales al profesor.

### 5.3. Sobre el informe

Los estudiantes deberán presentar un informe en formato *Markdown* donde se las siguientes características del diseño final (el archivo se cargará como archivo `README.md` del repositorio):

1. Una descripción general del funcionamiento del circuito completo y de cada subsistema.
2. Diagramas de bloques de cada subsistema y su funcionamiento fundamental, según descritos en la sección 5.
3. Diagramas de estado de todas las FSM diseñadas (si existen), según descritos en la sección 5.
4. Ejemplo y análisis de una simulación funcional del sistema completo, desde el estímulo de entrada hasta el manejo de los 7 segmentos.
5. Análisis de consumo de recursos en la FPGA (LUTs, FFs, etc.) y del consumo de potencia que reporta la herramienta Vivado.
6. Reporte de velocidades máximas de reloj posibles en el diseño (mínima frecuencia de reloj para este diseño: 50 MHz).
7. Análisis de principales problemas hallados durante el trabajo y de las soluciones aplicadas.

## 6. Evaluación (100 %)

Rubro	Fecha de Entrega Límite	Valor (%)	Detalles
Propuesta de plan de trabajo	22 de setiembre	15	Se revisará un esquema de plan de trabajo con cada grupo personalmente sobre la plataforma del repositorio en GitHub ( ver). Debe indicar quienes serán los integrantes responsables para cada tarea y cuales son sus especificaciones de forma clara (entradas, salidas, casos de prueba). Además se revisará la correcta instalación de Vivado (o ISE para quienes usen tarjetas Nexys 3) en sus máquinas de trabajo.
Avance	4 de octubre	35	Se revisará el cumplimiento del plan de trabajo realizado hasta el momento y se verificará al menos el funcionamiento en simulación RTL como mínimo (se asignarán citas para el grupo del profesor Chacón Rodríguez para este fin). Problemas encontrados durante el transcurso del proyecto deben ser reportados en el plan de trabajo.
Defensa del proyecto	11 de octubre	50	Cada grupo realizará una demostración del funcionamiento del diseño implementado en una placa de FPGA (Nexys 3, Nexys 4, Basys 3 o Nexys A7). se asignarán citas para el grupo del profesor Chacón Rodríguez para este fin. Se revisará el cumplimiento de la especificación solicitada y del plan de trabajo. La mitad de este 50 % dependerá del informe adjunto que se monte en el repositorio (en el archivo README.md del proyecto, usando formato <i>Markdown</i> ).

## Referencias

- [1] Digilent. *Basys 3 Artix-7 FPGA Board Reference Manual*. URL: [https://digilent.com/reference/\\_media/reference/programmable-logic/basys-3/basys3\\_rm.pdf](https://digilent.com/reference/_media/reference/programmable-logic/basys-3/basys3_rm.pdf).
- [2] Digilent. *Nexys 3<sup>TM</sup> FPGA Board Reference Manual*. URL: [https://digilent.com/reference/\\_media/nexys:nexys3:nexys3\\_rm.pdf](https://digilent.com/reference/_media/nexys:nexys3:nexys3_rm.pdf).
- [3] Digilent. *Nexys 4<sup>TM</sup> FPGA Board Reference Manual*. URL: [https://digilent.com/reference/\\_media/reference/programmable-logic/nexys-4/nexys4\\_rm.pdf](https://digilent.com/reference/_media/reference/programmable-logic/nexys-4/nexys4_rm.pdf).
- [4] Digilent. *Nexys A7 FPGA Board Reference Manual*. URL: [https://digilent.com/reference/\\_media/reference/programmable-logic/nexys-a7/nexys-a7\\_rm.pdf](https://digilent.com/reference/_media/reference/programmable-logic/nexys-a7/nexys-a7_rm.pdf).

- [5] Digilent. *Video tutorial para principiantes. Vivado y Basys 3*. URL: [https://www.youtube.com/watch?v=t0wMmBI\\_XNo](https://www.youtube.com/watch?v=t0wMmBI_XNo).
- [6] Whitney Knitter. *Getting Started with FPGA Design #1: Installing Xilinx Vivado/Vitis*. URL: <https://www.youtube.com/watch?v=or3yYwGyGpA>.