



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

IIC2343 - Arquitectura de Computadores (II/2019)

Entrega 1-D

Entrega: 16 de Septiembre de 2019 | 10:59:59 a.m.

Requisitos

- Esta entrega es estrictamente individual. Cualquier tipo de falta a la [honestidad académica](#) será sancionada con la **reprobación** del curso con la nota mínima.
- Los nombre de archivos y el cómo deben ser ejecutados son parte del formato, no respetarlo será penalizado.
- El programa de la **placa** deberá ser realizado en [VHDL](#).
- La **documentación** deberá ser realizada en un archivo [Markdown](#) y subirlo junto a su tarea, de nombre [README.md](#), en el mismo repositorio.
- Esta entrega deberá ser subida a su repositorio personal de [GitHub](#) correspondiente en la fecha y hora dada.
- La entrega de la placa debe ser realizada previa o al inicio de la hora de ayudantía del curso. El no cumplimiento, no solo perjudicará su nota, sino también a sus compañeros

Introducción

La máquina de café del DCC, la DCCoffeeMachine se descompuso y esta en búsqueda de alguien que la repare, para eso, deberás usar tus conocimientos en sistemas secuenciales y rearmar el sistema de conteo de monedas.

Deberás crear un registro que cuente las monedas que lleva ingresadas en la máquina, debido a otro desperfecto, solo funciona con monedas de 100, y no recibe más de 700 que es el precio del café más caro.

Misión

Deberá crear un módulo VHDL que represente el contador de monedas, que **reciba una moneda con el boton UP** y que encienda los **leds del display 7 segmentos con el dinero ingresado hasta el momento**.

- Si ya hay 700 en la máquina y presiono el botón, la máquina entrega el café y vuelve a estar en 0.

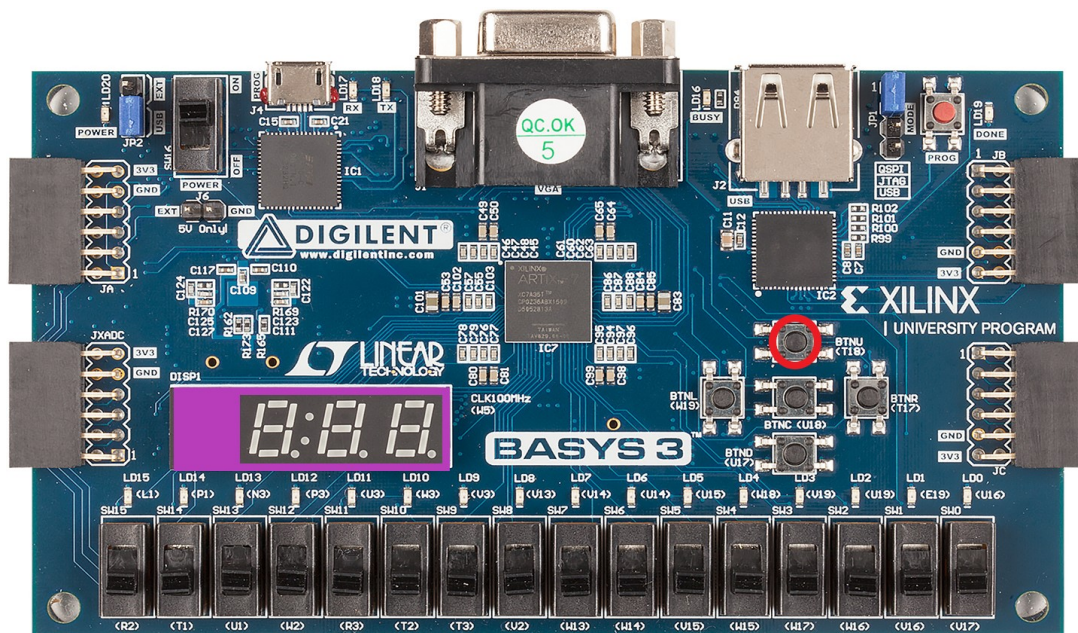
Se te entregará el módulo `Display_Controller.vhd` que puedes usar de manera opcional.

Se te entregará el módulo `Debouncer.vhd` que deberás usar para que los botones funcionen de manera correcta.

Debe **documentar con lo solicitado en el README.md**.

Placa

Utilizarán la placa Digilent Basys3 presente en la imagen, de la cual, usando el **boton UP** como input de la moneda, por otra parte, el output debe ser representado en los **display de 7-segmentos**.



Módulos entregados

Se le entrega un módulo VHDL para facilitar su entrega, este puede ser utilizado de **manera opcional**:

- `Display_Controller.vhd`

Este módulo recibe 4 vectores y un valor lógico: `dis_a[3:0]`, `dis_b[3:0]`, `dis_c[3:0]`, `dis_d[3:0]` y `clk`. Entrega como salida los vectores `seg[7:0]` y `an[3:0]`.

Su función es decodificar los valores que quieres entregar en los display 7 segmentos correspondientes:



Para que este módulo funcione, debe conectar la señal de entrada de la Basys3 `clk`, a la señal de entrada `clk` de este módulo.

La salida de este módulo genera las señales de salida de la Basys3: `seg[7:0]` y `an[3:0]` para encender el display.

Por otra parte se le entregará un módulo VHDL que deberá usar de **manera obligatoria**:

- `Debouncer.vhd`

Este módulo permite el correcto funcionamiento de los botones, dado que estos generan una señal vibratoria al ser pulsados.

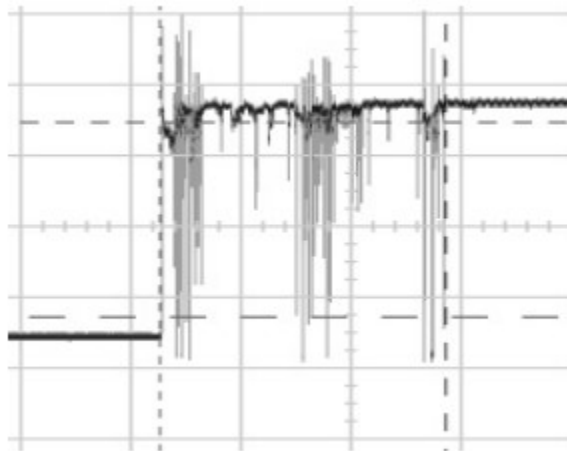


Figura 1: Señal al pulsar un botón

Recibe la señal de un botón y entrega la misma señal corregida.

Para que este módulo funcione, debe conectar la señal de entrada de la Basys3 `clk`, a la señal de entrada `clk` de este módulo.

Requerimientos

- Crea el proyecto
 - Selecciona las opciones correctas para crear el proyecto en Vivado, que funcione con la placa correspondiente.
 - Importa correctamente el archivo `Basys3.xdc` .
 - Configura correctamente las *constraints* del archivo `Basys3.xdc`. Descomentando las líneas correctas del archivo.
- Crea el módulo `dccoffee_machine.vhd`
 - Crear una source llamada `dccoffee_machine`, que contiene la arquitectura requerida para resolver el problema.
 - Puede crear más *sources* para facilitar el problema
- Incluir el `README.md` con lo solicitado.
- Entrega inmediata de la placa la siguiente semana

Entrega

La entrega se realizará a través de GitHub. El repositorio debe contener una carpeta con su proyecto de Vivado y el archivo `.bit`. En el caso de la carpeta del proyecto, deben subir solo la carpeta `.srcs`, el archivo `.xpr` y el archivo `Basys3.xdc`