



# Proyecto 1

## Primer Proyecto Zybo Z7

---

### 1. Instrucciones

Para cumplir con el objetivo del proyecto los alumnos deberán subir al repositorio el Wrapper generado en la plataforma de Vivado y la carpeta del proyecto generada en Vitis. Además, se deberá entregar un archivo `readme.txt` con las indicaciones y supuestos realizados para llevar a cabo la implementación en código C. Les pedimos por favor ser claros en este archivo, que le permitirá a los ayudantes poder corregir su trabajo de mejor forma, lo que directamente impactará en su calificación.

La lógica de programación empleada será libre. Es decir, podrán hacer uso de cualquier herramienta aprendida durante las clases, las tareas de programación en C e incluso alguna herramienta extra que deberán explicar en el archivo `readme.txt`.

Este trabajo deberá ser realizado en parejas y tanto la presentación como el código se entregarán a través de su repositorio de GitHub, respetando las siguientes fechas límites.

- Proyecto Implementado: **viernes 21 de Octubre del 2022, hasta las 23:59.**
- Presentación: **lunes 24 de Octubre del 2022**, en horario de clases.

### 2. Objetivo

Al finalizar este trabajo el estudiante será capaz de desarrollar la estructura necesaria para poder implementar GPIO, interrupciones, timers y protocolo de comunicación UART en la Zybo Z7. Además, será capaz de desarrollar código C de manera eficiente empleando los conceptos adquiridos en clases y en tareas anteriores.

### 3. Simón Dice

En este primer proyecto usaremos los conocimientos básicos del juego Simón Dice. La idea es emplear los 4 LEDs de la Zybo Z7 para generar progresivamente una secuencia aleatoria

de encendido y poder seguir esta secuencia a través de los 4 botones. Es decir, los 4 LEDs serán las instrucciones y los 4 botones serán las respuestas del usuario. Para llevar a cabo esto se requerirá cumplir con las funcionalidades descritas en la Sección 3.2.

### 3.1. Flujo del programa

- En primer lugar se deberá implementar el Switch 0 (SW0) para que permita encender el juego (UN SWITCH NO UN BOTÓN). Esta implementación no debe apagar la Zybo.
- Una vez encendido el juego, se desplegará en la consola del computador un mensaje solicitando al jugador que ingrese su nombre. Se deberá verificar que no existe dicho nombre, si es que existe el nombre se le deberá preguntar al jugador si es la misma persona, en el caso de que no sea la misma persona tendrá que ingresar otro nombre.
- Una vez ingresado el nombre, se desplegará un menú controlado por los botones, el cual deberá tener 2 opciones. La primera corresponde a jugar Simón Dice y la segunda corresponde a mostrar los mejores puntajes obtenidos mientras la Zybo siga encendida.
- Luego, para comenzar el juego se deberá presionar el botón indicado en el menú anterior. Para señalar que el juego comenzará, se deberán hacer parpadear los cuatro LEDs por 2 segundos, con una frecuencia de medio segundo. (es decir on-off-on-off y luego comenzar con la secuencia).
- Cada ronda de juego constará de dos partes: (1) generación y reproducción de secuencia y (2) repetición por parte del jugador:
  - **Generación y reproducción de secuencia:** se irá generando una secuencia de forma progresiva, es decir, se reproducirá toda la secuencia de LEDs de la ronda anterior, más un LED aleatorio. Así, en la primera ronda se encenderá un LED aleatorio, luego, en la segunda ronda, se encenderá dicho LED más otro adicional aleatorio (puede ser el mismo) y así sucesivamente, donde en cada ronda, la secuencia se irá haciendo más larga.
  - **Repetición por parte del jugador:** Una vez que se genere y reproduzca una secuencia, el jugador tendrá que presionar los botones de forma secuencial, repitiendo la secuencia mostrada.
- El juego podrá terminar de dos formas: (1) el jugador se equivoca al momento de repetir la secuencia o (2) pasan 30 segundos desde que se apretó el último botón (en el caso de que ya haya repetido parte de la secuencia) o 30 segundos desde que empezó la ronda (en el caso en que no haya presionado ningún botón), es decir, 30 segundos inactivos. NOTA: NO se cuenta el tiempo en el cual la Zybo está reproduciendo la secuencia, sino que solo se cuenta el tiempo inactivo dentro de la ronda. No se apure en presionar los LEDs. Este es un juego de memoria, no es un juego de velocidad, por lo que repetir la secuencia rápido no importa.

- Una vez que se pierde, los 4 LEDs tendrán que parpadear 3 veces para indicar que se terminó el juego. Frecuencia de medio segundo, es decir on-off-on-off-on-off.
- La zybo deberá enviar a la consola del computador el tiempo de juego y la cantidad de rondas completadas. Los 3 mejores valores tendrán que ser guardados en una estructura.
- Una vez terminado el juego de un jugador, el PC deberá preguntar al usuario si este desea volver a jugar o si otro usuario quiere jugar. En caso de que el mismo jugador juegue nuevamente, el PC solo guardará su puntaje más alto. En caso que otro usuario entre en juego, este deberá ingresar su nombre. El PC deberá ser capaz de reconocer si ese nombre ya existe en la estructura y consultar al usuario si es que ese es su nombre, en caso de que no, el usuario deberá escoger otro nombre. Para realizar esto, deberán usar las funciones *malloc* y *realloc* para reasignar memoria a medida que el número de usuarios aumenta.

### 3.2. Vivado

Como se ha visto en los laboratorios, para poder acceder al uso de la plataforma Vitis e implementar código en C siempre es necesario pasar por Vivado y generar la arquitectura que se desea usar. Completar esta sección no tiene puntaje dentro de la evaluación. Sin embargo, se penalizará a aquellos grupos que no logren explicar el funcionamiento e implementación de esta sección.

Para cumplir con el objetivo esperado de esta sección el alumno debe lograr generar una estructura que permita el uso de:

- **GPIOs:** Se requerirán 4 LEDs, 4 Botones y 1 Switch disponible como entrada de la Zybo Z7. Recuerde que la Zybo debe mantenerse encendida en todo momento.
- **UART:** Cumplirá el rol de comunicar el PC y la Zybo Z7.
- **Timers:** Pueden ser activados los timers que se deseen.
- **Interrupciones:** Se requerirán las interrupciones de los botones y de los timers.

Finalmente se deberá exportar el proyecto a la plataforma Vitis, para poder emplear la programación en C y desarrollar las funcionalidades de la tarjeta.

### 3.3. Vitis

Como ya se mencionó la forma de implementar esto será libre. Sin embargo, se pedirán algunos requisitos mínimos, los cuales se mencionan a continuación:

### 3.3.1. Uso de GPIOs

- Se deberá tener un switch (SW0) que permita encender el juego. Es decir, al mover el switch se le permitirá jugar al usuario, en caso contrario no se podrá acceder a la etapa de juego. Recuerde que esta implementación no debe apagar la Zybo.
- El encendido y apagado de los 4 LEDs deberá ser implementado mediante timers. Así, para la «generación y reproducción de secuencia» por parte de la Zybo, deberán mantener encendido el LED por 0.8 segundos y apagarlo por 0.2 segundos, para luego encender el siguiente bajo la misma regla.
- El estado de los 4 botones se deberá leer mediante interrupciones.
- En caso de que el operador se encuentre en el menú los botones deben permitir seleccionar las opciones disponibles. Es decir, en consola deben imprimirse las opciones con el botón que se deba presionar para seleccionarla (Ejemplo: «*Presione (BTN0) para JUGAR*»)

### 3.3.2. UART

- Cuando el juego este apagado (switch en OFF), se deberá imprimir 1 sola vez las instrucciones para poder encender la consola.
- Cuando el juego se encienda se deberá transmitir mensaje de bienvenida al usuario.
- Una vez que el juego este encendido se deberá poder ingresar el nombre del usuario y se deberá generar un menú controlado por los botones, el cual deberá tener 2 opciones. La primera corresponde a jugar Simón Dice y la segunda corresponde a mostrar los 3 mejores puntajes obtenidos mientras la Zybo siga encendida.
- La interfaz que interactúe con el usuario debe ser cómoda e intuitiva. No es necesario realizar algo tan complejo, basta con instrucciones dentro de una consola.

### 3.3.3. Timers

- Como se indicó en la sección de «Uso de GPIOs» el/los timer/s deberá/n llevar el conteo del tiempo que transcurre entre los encendidos de los LEDs.

### 3.3.4. Interrupciones

- La lectura de los botones deberá ser implementada a través de interrupciones.
- El funcionamiento del timer deberá ser realizado a través de interrupciones.

## 4. Reglas Generales

Para llevar a cabo el juego, se deberán tener en consideración las siguientes instrucciones adicionales.

- El juego debe ser capaz de almacenar los mejores 3 puntajes mientras la Zybo siga encendida.
- El jugador pierde si se equivoca al repetir la secuencia o si permanece inactivo durante 30 segundos en una ronda.
- La única forma de volver al menú principal es apagar y prender el juego (switch en OFF y ON, no la Zybo) o perder según las reglas del punto anterior.
- El puntaje debe ser contabilizado a partir de la cantidad de rondas que complete el jugador. Es decir, si logra completar 10 rondas sin perder y pierde en la ronda N°11, el puntaje asignado es 10.

## 5. Recomendaciones

Dado que es un trabajo en parejas, parte de la complejidad radica en que ambos entiendan el código que escriben. De esta forma, se les recomienda leer la guía de estilo disponible en la plataforma del curso, para así generar un estándar con su compañero y poder avanzar de la forma más ordenada posible.