

PRÁCTICA 3 DESCRIPCIÓN DE HARDWARE (RTL)

El objetivo de esta práctica es diseñar un circuito digital para jugar a la ruleta. El sistema contará con 3 botones **APOSTAR**, **GIRAR** y **PARAR** (para apostar, hacer girar la ruleta y detener el giro respectivamente) y 4 displays 7 segmentos, uno para visualizar el valor actual de la ruleta, otro con la apuesta, y los últimos para visualizar el dinero disponible.

IMPORTANTE: Para el bloque de datos de los diseños RTL utilizar flip flops tipo D con entrada enable (primitiva DFFE de Quartus) conectando a la misma las señales CSL del bloque de control. En todos los casos utilizar flip flops sensibles al flanco de subida.



El dinero disponible se va a representar con números BCD y se visualizará en los display 2 y 3.

La apuesta y la ruleta tomarán 6 valores posibles (segmentos a - f). La apuesta se indicará en el display 1 y la ruleta en el display 0.

La forma de jugar será la siguiente:

Inicialmente se dispondrá de \$5 que se indicarán en el display 2 y la apuesta indicará solo el segmento “a” del display 1 encendido. La ruleta también inicializará solo con el segmento “a” encendido.

Antes de largar a girar la ruleta, se deberá elegir un segmento a apostar. Manteniendo oprimido el botón *Apostar* irá “girando lentamente” el segmento de apuesta, de forma que se podrá detener en el lugar deseado (el segmento a apostar).

Una vez hecha la apuesta, se debe oprimir el botón *Girar*, lo que provocará que la ruleta “gire” muy rápido y seguidamente el botón *Parar*, lo que detendrá el giro de la ruleta.

Según los resultados puede suceder lo siguiente:

- Si la posición final de la ruleta coincide con la de la apuesta, se incrementa el dinero disponible en \$3.
- Si la posición final de la ruleta no coincide con la de la apuesta, se decrementa el dinero disponible en \$1.

El juego finalizará cuando se llega a \$0 o se supera los \$9, quedando los botones inactivos (el sistema hará caso omiso a cualquier botón). Para volver a jugar se deberá resetear el sistema.

El jugador perderá si llega a \$0, lo que se indicará con un 0 en el display 2.

El jugador ganará si se supera los \$9, lo que se indicará prendiendo el Led 1, y en los display 2 y 3 se indicará el monto total ganado, ver parte d de la práctica 1.

Durante el juego el led permanecerá apagado.

Parte a:

En esta parte se verá un circuito que solo permitirá apostar y hacer girar la ruleta, no manejará dinero, indicando con una salida *Gane* si se ganó o no en la última jugada.

La siguiente descripción RTL, corresponde al circuito **RTL_A**, que comanda al bloque “**Apuesta**” (ver figura 1). Como bloque “**Apuesta**” debe utilizarse el bloque “**Contador_6**” diseñado en la práctica 2.

Inputs: Apostar; Girar; Parar; Apuesta[6]; Din_ini[4].
 Outputs: Ruleta[6]; Dinero[4]; Ena_ap; Gane.
 Memory: Din[4]; Rul[6]; G.

0. $Din \leftarrow Din_ini$
 $Rul \leftarrow 111110b$
 $G \leftarrow 0;$
 $Ena_ap = 1;$
1. $Ena_ap = Apostar$
 $\rightarrow (Girar, /Girar) / (1,2)$
2. $Rul * Parar \leftarrow Rul[4:0], Rul[5]$
 $G * /Parar \leftarrow Win$
 $Ena_ap = 1;$
 $\rightarrow (Parar, /Parar) / (2,1)$

End Sequence

Control Reset (0)

$Win = Igual(Apuesta, Ruleta)$
 $Ruleta = Rul$
 $Gane = G$
 $Dinero = Din$

End

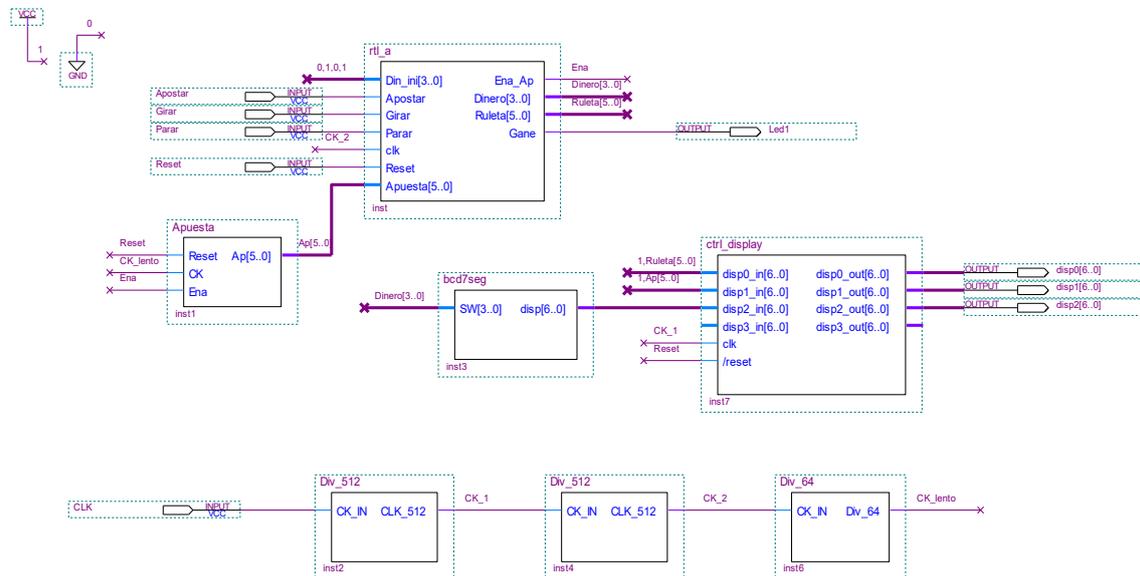


Figura 1. Circuito de prueba de la parte a).

Los bloques **DIV_512**, **DIV_64**, **ctrl_display** y **BCD7SEG** corresponden a los utilizados en las prácticas anteriores.

Se pide:

- Analizar la descripción RTL dada.
- Generar los proyectos **RTL_A_datos** y **RTL_A_control** con los diseños correspondientes a los bloques de datos y control de **RTL_A**.
- Crear los bloques **RTL_A_datos** y **RTL_A_control**.
- Generar un proyecto **RTL_A** a partir de los bloques definidos anteriormente. Simular.
- Crear el bloque **RTL_A**.
- Generar un proyecto con el circuito de la figura 1, donde *RESET* se conectará al switch SW1, *Apostar*, *Girar* y *Parar* a los pulsadores boton1, boton2 y boton3 y *CLK* al reloj de la placa DE0. Las salidas se conectarán al led 1 y al display 7 segmentos como es habitual.
- Analizar lo que se deberá observar y el por qué, simular, programar la placa y verificar el funcionamiento (para evitar tiempos largos en la simulación, realizarla sin los divisores).

Parte b:

En esta parte se debe incorporar el manejo del dinero en cada apuesta.

Para esto, modificar el valor del registro **Din** en el paso 2 al oprimir el botón *Parar*, según el siguiente detalle:

- Si se ganó la tirada (Apuesta = Ruleta) entonces se suma 3 al registro Din.
- Si se perdió la tirada (Apuesta <> Ruleta) entonces se resta 1 al registro Din.

Para sumar y restar, se debe utilizar el bloque **Suma_bcd** diseñado en la práctica 1. Notar que el resultado de restar 1 es el mismo que sumar 9.

Tener en cuenta que la transferencia no debe realizarse en todos los períodos de reloj en que la ruleta se encuentra girando, sino que lo debe hacer cuando la ruleta se detuvo (al oprimir *Parar*).

En esta versión, se eliminará la salida *Gane* y no se contemplarán los casos en que haya un desborde de dinero (= \$0 o >\$9).

Se pide:

- Modificar la descripción **RTL_A** para agregarle las funcionalidades especificadas.
- Generar los proyectos **RTL_B_datos** y **RTL_B_control** con los diseños correspondientes a los bloques de datos y control de **RTL_B**.
- Crear los bloques **RTL_B_datos** y **RTL_B_control**.
- Generar un proyecto **RTL_B** a partir de los bloques definidos anteriormente. Simular.
- Crear el bloque **RTL_B**.
- Generar un proyecto con un circuito similar al de la figura 1, con la diferencia que se sustituye **RTL_A** por **RTL_B**, y se anula la salida Led1. En este nuevo circuito *RESET* se conectará al switch SW1, *Apostar*, *Girar* y *Parar* a los pulsadores boton1, boton2 y boton3 y *CLK* al reloj de la placa DE0 y las salidas al display 7 segmentos como es habitual.
- Analizar lo que se deberá observar y el por qué, simular, programar la placa y verificar el funcionamiento (para evitar tiempos largos en la simulación, realizarla sin los divisores).

Parte c:

En esta parte se agregará más funcionalidad para indicar si el jugador ganó el juego (superó los \$9) o lo perdió (llegó a \$0).

En cada tirada, al oprimirse el botón *Parar*, en el paso 2, se debe tener presente:

- Si el valor de Din = 1 y se pierde la tirada, entonces el jugador pierde, finalizando el juego.
- Si (Din + 3) tiene carry y se gana la tirada, entonces el jugador gana, finalizando el juego.
- En otro caso, el juego continúa, permitiendo una nueva apuesta.

Una vez finalizado el juego, el circuito deberá quedar “bloqueado” haciendo caso omiso a cualquier botón. Para volver a jugar, se deberá resetear el sistema.

Además, se debe agregar la salida *Gane* la cual se pondrá a 1 si finaliza el juego y el jugador gana. Se sugiere utilizar el carry de Din + 3 convenientemente para obtener el valor de la salida *Gane*.

Tener presente que el jugador perderá si llega a \$0, lo que se debe indicar con un 0 en el display 2; el jugador ganará si se supera los \$9, lo que se indicará prendiendo el Led 1, y en los display 2 y 3 se indicará el monto total ganado, ver parte d de la práctica 1. Durante el juego el led permanecerá apagado.

Se pide:

- Modificar la descripción **RTL_B** para agregarle las funcionalidades especificadas.
- Generar los proyectos **RTL_C_datos** y **RTL_C_control** con los diseños correspondientes a los bloques de datos y control de **RTL_C**.
- Crear los bloques **RTL_C_datos** y **RTL_C_control**.
- Generar un proyecto **RTL_C** a partir de los bloques definidos anteriormente. Simular.
- Crear el bloque **RTL_C**.
- Generar un proyecto con un circuito similar al de la figura 1, con la diferencia que se sustituye **RTL_A** por **RTL_C**. En este nuevo circuito **RESET** se conectará al switch SW1, *Apostar*, *Girar* y *Parar* a los pulsadores boton1, boton2 y boton3 y **CLK** al reloj de la placa DE0 y las salidas al display 7 segmentos como es habitual.
- Analizar lo que se deberá observar y el por qué, simular, programar la placa y verificar el funcionamiento (para evitar tiempos largos en la simulación, realizarla sin los divisores).

NOTA: Para poder forzar la terminación del juego se sugiere conectar el reloj *CK_lento* en el bloque **RTL_C** para que la ruleta gire lentamente y así poder pararla en una posición deseada. De esta forma se pueden generar apuestas ganadoras o perdedoras a gusto.

INFORME

Uno de los integrantes del grupo deberá entregar en la tarea correspondiente en EVA el **informe.pdf** conteniendo:

Parte a)

- Diagrama de los circuitos **RTL_A_Control**, **RTL_A_Datos** y **RTL_A**.
- Breve descripción de lo que hace **RTL_A**.
- Simulaciones en donde se detallen los cambios de pasos y valores de los registros. Se deben incluir también los instantes en que se oprimen y sueltan los botones *Apostar*, *Girar* y *Parar*.
- Breve explicación de cómo funciona el circuito de la figura 1 y lo que se va a ver en la placa.

Parte b)

- Descripción RTL de **RTL_B**.
- Breve descripción de la funcionalidad incorporada.
- Diagrama de los circuitos **RTL_B_Control**, **RTL_B_Datos** y **RTL_B**.
- Simulaciones en que se detalle fundamentalmente la nueva funcionalidad.

Parte c)

- Descripción RTL de **RTL_C**.
- Breve descripción de la funcionalidad incorporada.
- Diagrama de los circuitos **RTL_C_Control**, **RTL_C_Datos** y **RTL_C**.
- Simulaciones en que se detalle fundamentalmente la nueva funcionalidad.

El día de la evaluación **el grupo deberá presentarse puntualmente 10 minutos antes de la hora establecida** en el laboratorio de software del IIE.

Además se deberán traer el KIT DE0, la letra de la práctica y un “pen drive usb” con **todos los archivos de todos los proyectos** indicados y las simulaciones realizadas.

Los proyectos deben tener los pines asignados y deben estar compilados, prontos para ser probados en la placa DE0. Durante la evaluación **NO** se dispondrá de tiempo para realizar estas tareas, modificar el circuito ni generar nuevas simulaciones.

Puede ser de utilidad: Para “pegar” en el informe un circuito realizado en el editor gráfico del Quartus se debe seleccionar el circuito, “copiarlo” y realizar un “pegado especial”, indicando “mapa de bits”.

Notación:

Para la descripción RTL, si se está utilizando un bloque que tiene, por ejemplo, 3 entradas (A, B y Cyin) y 2 salidas (Suma y Cyout), por simplicidad, notar cada salida por separado, tal como se indica en el siguiente ejemplo:

```

.....
Registro1 ← Suma (A,B,Cyin)
Registro 2 ← Cyout (A,B,Cyin)
.....

```