

PRÁCTICA 3

DESCRIPCIÓN DE HARDWARE (RTL)

El objetivo de esta práctica es diseñar un circuito digital que simule el funcionamiento de un control remoto diseñado con el fin de moverse dentro de distintos documentales o episodios de una serie.

El sistema contará con 3 entradas: 1 switch, **PP** (Play/Pause) y 2 botones, **FF** y **REW** (Forward y Rewind); y los 4 displays 7 segmentos como salida, dos para mostrar la entrada seleccionada, uno para visualizar el episodio que se está reproduciendo y el otro para indicar el estado de la reproducción en cada momento (play, forward, rewind o pause).



Las entradas seleccionadas se mostrarán en los displays 3 y 2 utilizando el circuito **disp_comand** diseñado en la primera práctica del curso.

El número de pista se va a representar con números BCD y se visualizará en el display 1.

El estado de la reproducción se indicará en el display 0 con la ilusión de giro de un segmento (similar a lo realizado en la primera parte de la práctica 2) según la siguiente descripción:

- *Play* (reproducción): el segmento girará en sentido horario a aproximadamente 1 revolución por segundo.
- *Forward* (avance): el segmento girará en sentido horario a aproximadamente 2 revoluciones por segundo (2 veces más rápido que en estado de *Play*).
- *Rewind* (retroceso): ídem Forward pero en sentido antihorario.
- *Pause* (pausa): el segmento permanecerá inmóvil.

Las funciones de cada entrada serán las siguientes:

- Si el switch *PP* está en 0 se indicará “*Pausa*” y en caso contrario se indicará “*Play*” (reproducción).
- Si se oprime *FF* un tiempo “corto” se deberá pasar a la próxima pista (salvo que la actual sea la 8) e inicializar en “b” el segmento que rota.
- Si se oprime *FF* un tiempo “largo” se deberá “avanzar” hasta que se suelte *FF*.
- Si se oprime *REW* un tiempo “corto” se deberá pasar a la pista anterior (salvo que la actual sea 1) e inicializar en “f” el segmento que rota.
- Si se oprime *REW* un tiempo “largo” se deberá “retroceder” hasta que se suelte *REW*.
- En estado de *Pausa*, se hará caso omiso a los botones *FF* y *REW*.
- En estado de avance se hará caso omiso a los botones *REW* y *PP*.
- Una vez en retroceso se hará caso omiso a los botones *FF* y *PP*.

El circuito **CortoLargo** diseñado en la práctica 2 se utilizará para dar doble funcionalidad a los botones *FF* y *REW*, y así diferenciar si fueron oprimidos un tiempo “corto” o un tiempo “largo”.

Luego de un Reset, el circuito deberá indicar el episodio 1 en el display 1 y en el display 0 deberá inicializarse con el segmento “a”. Luego deberá pasar a estado de pausa y según la entrada *PP* permanecer ahí o pasar al estado de reproducción.

El reloj para los circuitos RTL y modo reloj debe ser 11.92 Hz.

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.

Parte a:

En esta parte se verá un circuito que tendrá las funciones de Play, Pausa y avance. No se incluyen las funciones de pasar a la siguiente pista, ni las relacionadas con el retroceso. Dado que se considera solo el avance, se dispone de una entrada FF_L sin tener en cuenta el tiempo en que este botón es oprimido. Esta característica se introduce en la parte b). La siguiente descripción RTL, corresponde al circuito **RTL_A** de la figura 1.

Module: RTL_A
 Inputs: PP, FF_L
 Outputs: Pista, Posicion[6], Comando[2]
 Memory: pos[6], pst[4], Flag

- 0. $pst \leftarrow 1$
 $pos \leftarrow 111110b$
- 1. Comando= 11b
 $\rightarrow (/PP, PP) / (1,2)$
- 2. $pos * Flag \leftarrow pos[4..0], pos[5]$
 Comando= 00b
 $\rightarrow (/PP, PP \wedge FF_L, PP \wedge /FF_L) / (1,2,3)$
- 3. $pos \leftarrow pos[4..0], pos[5]$
 Comando= 01b
 $\rightarrow (FF_L, /FF_L) / (2, 3)$

End Sequence
 Control Reset (0)
 Posicion = pos
 Pista = pst
 Flag $\leftarrow /Flag$
 End

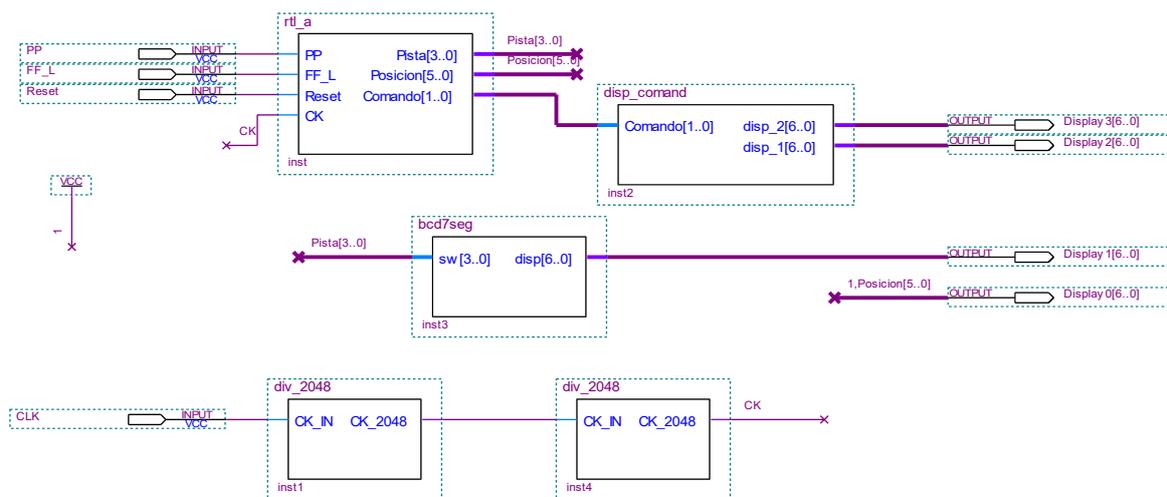


Figura 1. Circuito de prueba de la parte a).

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 SW9, **PP** al switch SW0, **FF_L** al pulsador 1 y **CLK** al reloj de la placa DE0. Las salidas se conectarán al display 7 segmentos como es habitual.
- Analizar qué se debería observar y por qué. Simular, programar la placa y verificar el funcionamiento (para evitar tiempos largos en la simulación, realizarla sin el bloque DIV_2048).

Parte b:

En esta parte se debe incorporar la doble funcionalidad en el botón FF utilizando el circuito **CortoLargo** diseñado en la práctica 2 según se detalla en la figura 2.

Se debe modificar la descripción RTL para incorporar la funcionalidad de saltar al próximo episodio, utilizando el bloque combinatorio **INC** diseñado en la práctica 1.

Para esto, se sugiere agregar un paso más (paso 4) en la descripción, en donde se incremente la pista y se inicialice el valor del segmento que rota. También se deberá modificar la bifurcación del paso 2 para ir al paso 3 o 4 según las entradas **FF_C** o **FF_L**.

Observar que las entradas **FF_L** y **FF_C**, son las salidas del bloque **CortoLargo** cuya entrada botón es **FF**. Esto nos garantiza que **NO** se van a activar ambas en forma simultánea (por construcción de **CortoLargo**). Esto se puede aprovechar para simplificar la lógica de la bifurcación.

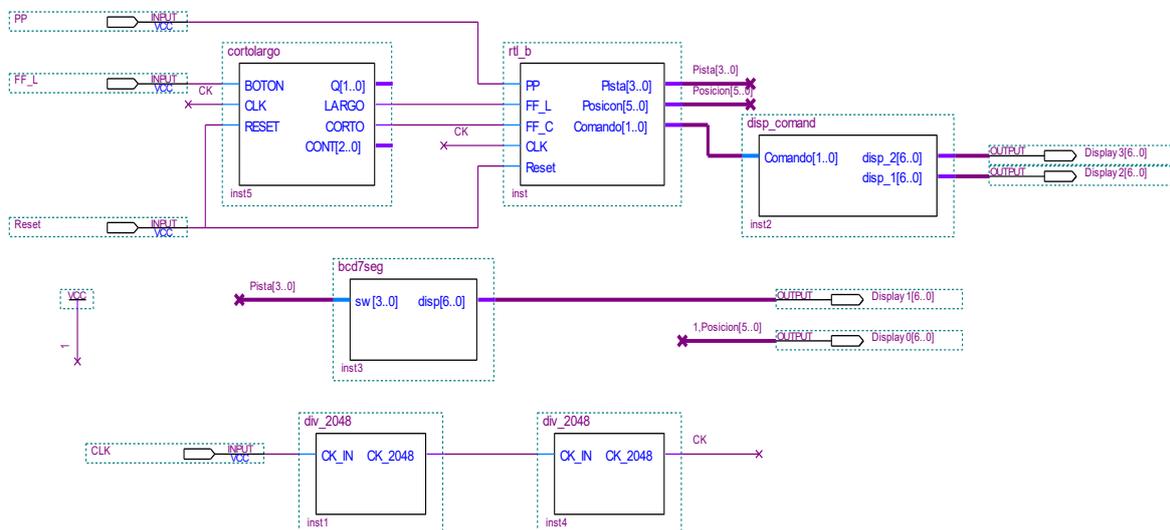


Figura 2. Circuito de prueba de la parte b)

Se pide:

- Modificar la descripción RTL de la parte 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 el circuito de la figura 2, donde **RESET** se conectará al switch SW9, **PP** al switch SW0, **FF** al pulsador 1 y **CLK** al reloj de la placa DE0. Las salidas se conectarán al display 7 segmentos como es habitual.
- Analizar qué se debería observar y por qué. Simular, programar la placa y verificar el funcionamiento (para evitar tiempos largos en la simulación, realizarla sin el bloque DIV_2048).

Parte c:

En esta parte, se debe diseñar un circuito **RTL_C** que funcione en forma idéntica al diseñado en la parte b), y que además incorpore las funciones de retroceso y salto al episodio anterior, teniendo en cuenta el mismo tipo de comportamiento que en avance. Para esto se deberán agregar las entradas **Rew_L** y **Rew_C** para cada función, respectivamente. En la figura 3 se puede ver cómo será utilizado **RTL_C**.

Se dispone del bloque combinatorio DEC, el cual decrementa un n° BCD a su salida y satura en 0.

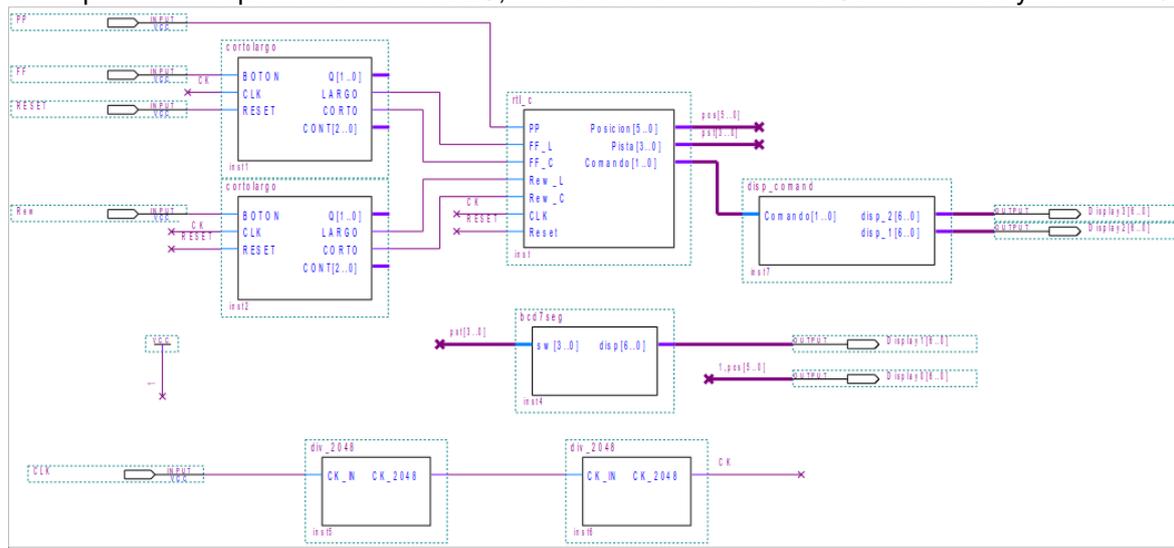


Figura 3. Circuito de prueba de la parte c)

Notas:

- Considerar para el diseño que los botones **FF** y **Rew** no serán oprimidos en forma simultánea.
- Dado que el decrementador satura en 0 y no en 1 (como es necesario), **esta funcionalidad deberá ser tomada en cuenta en la descripción RTL.**

Se pide:

- Realizar la descripción de **RTL_C**.
- 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 el circuito de la figura 3, donde *RESET* se conectará al switch SW9, *PP* al switch SW0, *FF* al pulsador 1, *Rew* al pulsador 2 y *CLK* al reloj de la placa DE0. Las salidas se conectarán al display 7 segmentos como es habitual.
- Analizar qué se debería observar y por qué. Simular, programar la placa y verificar el funcionamiento (para evitar tiempos largos en la simulación, realizarla sin el bloque DIV_2048).

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.
- 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_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_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, listos 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 combinatorio, que tiene, por ejemplo, 1 entrada (In) y 1 salidas (Out), notar su utilización tal como se indica en el siguiente ejemplo con el bloque INC:

.....
 Registro1 ← INC(A)

donde se explicita que el resultado de incrementar A se transfiere a Registro1.