

PRÁCTICA 3 DESCRIPCIÓN DE HARDWARE (RTL)

El objetivo de esta práctica es diseñar un circuito digital que implemente una versión del juego Master Mind en la placa DE0.

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 PASO del bloque de control. En todos los casos utilizar flip flops sensibles al flanco de subida.

En esta versión de Master Mind un jugador ingresa una clave de 4 dígitos, distintos entre sí, del 0 al 5. Una vez ingresada la clave, ésta se ocultará y entra al juego un segundo jugador. Este segundo jugador deberá descubrir la clave oculta ingresando adivinanzas de números de 4 dígitos, distintos entre sí, e irá recibiendo pistas de cuán cerca está de adivinar. El jugador tendrá un máximo de 8 intentos para adivinar la clave, en caso de que no adivine la clave en una cantidad de intentos menor o igual al máximo permitido, habrá perdido el juego.

Pistas:

- **Negras:** Son la cantidad de dígitos de la adivinanza que coinciden con un dígito de la clave y se encuentran en la posición correcta.
- **Blancas:** Son la cantidad de dígitos de la adivinanza que coinciden con un dígito de la clave pero no se encuentran en la posición correcta.

Sistema RTL:

En la figura al final del documento se puede ver cómo se conecta el sistema RTL a los bloques **CTRL_INCDEC**, **COMPARA_BN**, **SALIDA_LED** Y **7seg** previamente desarrollados. Se les entrega el bloque **ctrlEnter** que se conecta a un pulsador de la placa e indiferentemente del tiempo que el pulsador esté presionado genera un pulso de un período de reloj. Para utilizar el bloque **ctrlEnter** además de agregar el archivo **ctrlEnter** al proyecto se debe agregar los archivos **maq** y **div_2048**. Ver diseño de **ctrlEnter**.

Entradas RTL:

- **num[2..0]** – es la salida del bloque CTRL_INCDEC, la cual es un número de 3 bits entre 0 y 5 el cual utilizaremos para setear la clave y las distintas adivinanzas.
- **enter** – esta señal la utilizaremos para indicar que el valor de num[2..0] lo queremos setear en la clave o en la adivinanza, también para cuando visualizamos las pistas de nuestra adivinanza, que si todavía no terminó el juego, vamos a realizar otro intento.
- **negras[3..0]** – nos indica la cantidad de negras correspondientes a cierta adivinanza, se utilizará para saber cuándo se adivino la clave (4 negras) y en determinado momento para visualizar en los leds la cantidad de negras en la adivinanza.
- **blancas[3..0]** – nos indica la cantidad de blancas correspondientes a cierta adivinanza, se utilizará para en determinado momento visualizar en los leds la cantidad de blancas en la adivinanza.
- **clk** – señal de reloj de 50 MHz
- **/reset** – señal de reset activa por nivel bajo.

Salidas RTL:

- **LED_N[3..0]** – cantidad de negras para visualizar en los leds en código termómetro.
- **LED_B[3..0]** – cantidad de blancas para visualizar en los leds en código termómetro.
- **disp[2..0] (0,1,2y3)** – salida para visualizar tanto la clave como las adivinanzas en los displays
- **clave[2..0] (0,1,2y3)** – salida de los registros donde se guarda la clave para ingresar en COMPARA_BN y obtener las pistas.
- **adiv[2..0] (0,1,2y3)** – salida de los registros donde se guarda la adivinanza para ingresar en COMPARA_BN y obtener las pistas.
- **finPerdio** – salida que indica que se acabó el juego debido a que se agotaron los intentos y no se adivino la clave.
- **Clear** – salida para reiniciar el bloque CTRL_INCDEC.

Parte a) – Ingreso de números

En esta parte se verá un circuito que implementa el ingreso de números.

Tanto para ingresar los 4 dígitos de la clave como para ingresar los 4 dígitos de cada adivinanza, se irán entrando los dígitos de a uno por el display [0], empezando con el más significativo y se irán corriendo hacia la izquierda. Para seleccionar los dígitos se utilizarán los botones **inc** y **dec** y luego para ingresarlo el botón **enter**. A continuación se muestra un ejemplo de cómo se ingresa el número 1234: 0000 inc 0001 enter 0011 inc 0012 enter 0122 inc ... 1234 enter.

Para el bloque **INC** necesario en esta parte utilizar el bloque **INC** diseñado en la práctica 1 conectando el bit más significativo de la entrada a gnd.

MODULE ingresoNUM

Input: num[3]; enter; /reset.

Output: disp0[3]; disp1[3]; disp2[3]; disp3[3].

Memory: claveA[3]; claveB[3]; claveC[3]; claveD[3]; contPos[2]; contInt[3].

1 claveA ← 0; claveB ← 0; claveC ← 0
claveD ← 0; contPos ← 0; contInt ← 0

2 claveA*enter ← claveB
claveB*enter ← claveC
claveC*enter ← claveD
claveD*enter ← num
contPos*enter ← INC(contPos)

disp0 = num
disp1 = claveD
disp2 = claveC
disp3 = claveB

→ (pasar,!pasar)/(3,2)

3 disp0 = claveD
disp1 = claveC
disp2 = claveB
disp3 = claveA

→ 3

END SEQUENCE

Control reset(1)

pasar = contPos[1] and contPos[0] and enter

END

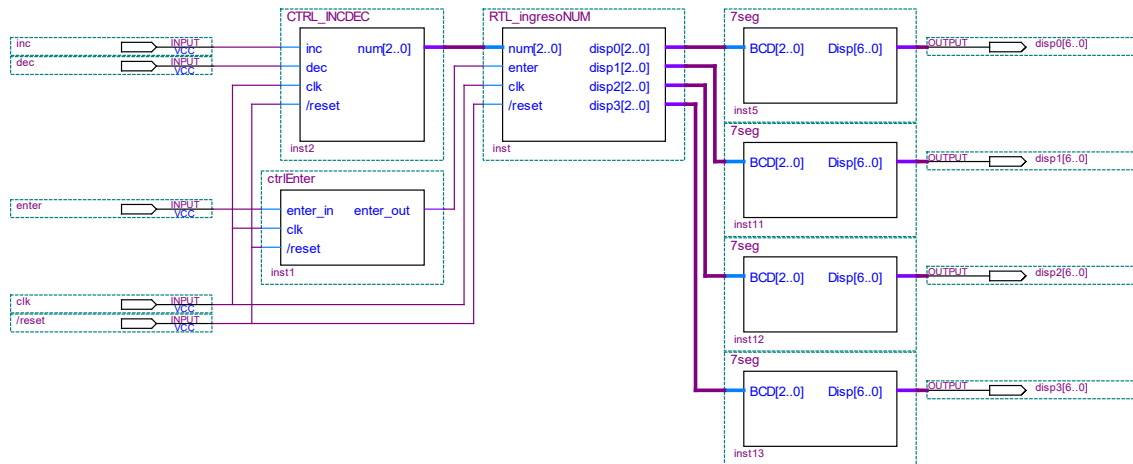


Figura 1.

Se pide:

- Analizar la descripción RTL dada y verificar que se realiza el ingreso de la clave de acuerdo a lo explicado anteriormente.
- Generar los proyectos **RTL_ingresoNUM_datos** y **RTL_ingresoNUM_control** con los diseños correspondientes a los bloques de datos y de control de **RTL_ingresoNUM**.
- Crear los bloques **RTL_ingresoNUM_datos** y **RTL_ingresoNUM_control**.
- Generar un proyecto **RTL_ingresoNUM** a partir de los bloques definidos anteriormente. Simular.
- Crear el bloque **RTL_ingresoNUM**.
- Simular el bloque **RTL_ingresoNUM**.
- Generar un proyecto con el circuito de la figura, las entradas **dec**, **inc** y **enter** irán asignadas a los pulsadores 2, 1 y 0 respectivamente, la entrada **/reset** será asignado al switch 0 y la entrada **clk** a la señal de reloj de 50 MHz de la placa DE0. Las salidas **disp0[6..0]**, **disp1[6..0]**, **disp2[6..0]** y **disp3[6..0]** a los display 0, 1, 2 y 3 respectivamente.
- Programar la placa y verificar el funcionamiento.

Parte b)

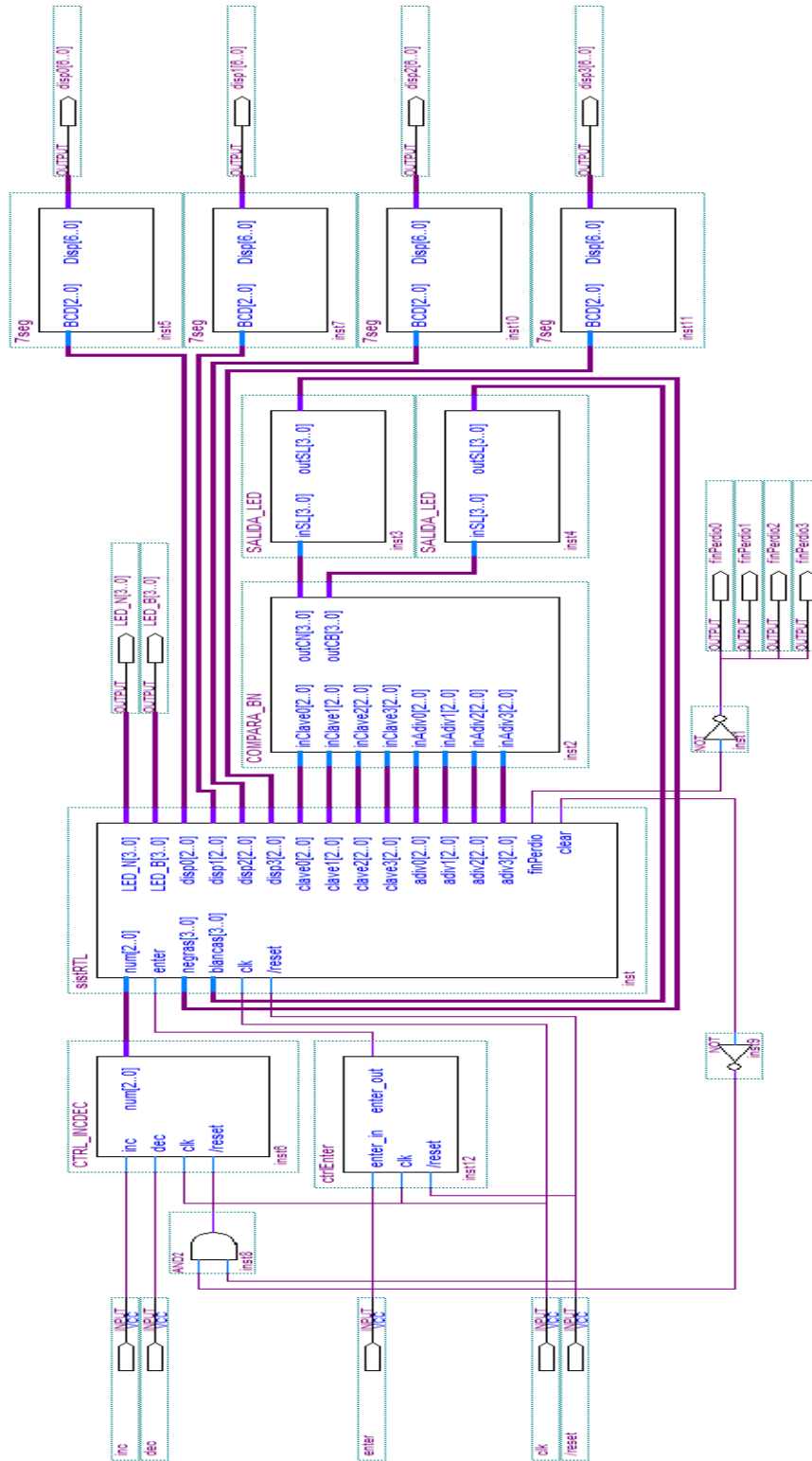
Ahora utilizando la parte a) se va a diseñar el sistema RTL descrito al principio del documento.

A tener en cuenta:

- Las adivinanzas y las claves se ingresan de igual manera como fue descrito en la parte a), se utiliza un contador de posiciones y el botón **enter** para saber cuándo se finaliza el ingreso de una adivinanza o la clave.
- Previo al ingreso de una nueva adivinanza se deben llevar los registros donde se almacena la misma a 0, y reiniciar el bloque **CTRL_INCDEC** con la señal **clear** (ver figura).
- Luego de ingresada la adivinanza, en caso de que el juego no haya finalizado, se debe visualizar la adivinanza en los displays y las pistas en los leds, y permanecer así hasta que el jugador presione **enter** para ingresar una nueva adivinanza.
- Similar al contador de posiciones se utilizará un contador de intentos para saber si el juego se da por perdido, para esto también utilizar el bloque **INC** diseñado en la práctica 1.
- Al finalizar el juego se debe visualizar la clave en los displays y las pistas en los leds, en caso de haber perdido se deben prender los puntos de los displays utilizando la señal **finPerdio** (ver figura). El juego se reinicia con la entrada **/reset**, de forma similar a la parte a).

Se pide:

- Realizar la descripción RTL.
- Generar los proyectos **sistRTL_datos** y **sistRTL_control** con los diseños correspondientes a los bloques de datos y control de **sistRTL**.
- Crear los bloques **sistRTL_datos** y **sistRTL_control**.
- Generar un proyecto **sistRTL** a partir de los bloques definidos anteriormente. Simular.
- Crear el bloque **sistRTL**.
- Simular el bloque **sistRTL**.
- Generar un proyecto **masterMind** con el circuito de la figura, las entradas **dec**, **inc** y **enter** irán asignadas a los pulsadores 2, 1 y 0 respectivamente, la entrada **/reset** será asignado al switch 0 y la entrada **clk** a la señal de reloj de 50 MHz de la placa DE0. Las salidas **disp0[6..0]**, **disp1[6..0]**, **disp2[6..0]** y **disp3[6..0]** a los display 0, 1, 2 y 3 respectivamente, las salidas **LED_N[3..0]** y **LED_B[3..0]** a los leds 9..6 y 3..0, y por último las salidas **finPerdio0**, **finPerdio1**, **finPerdio2** y **finPerdio3** será asignada a los cuatro puntos de los displays de la placa.
- Programar la placa y verificar el funcionamiento.



INFORME

Uno de los integrantes del grupo deberá entregar en la tarea correspondiente en EVA, un informe.pdf utilizando la carátula disponible en EVA, que incluya:

Parte a)

- Diagrama de los circuitos **RTL_ingresoNUM_Control**, **RTL_ingresoNUM_Datos** y **RTL_ingresoNUM**.
- Breve descripción de lo que hace **RTL_ingresoNUM**.
- Simulaciones en donde se detallen los cambios de pasos y valores de los registros.
- 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**.
- Breve descripción del funcionamiento.
- Diagrama de los circuitos **sistRTL_Control**, **sistRTL_Datos** y **sistRTL**.
- Simulaciones relevantes.
- Breve explicación de cómo funciona el circuito de la figura 2 y lo que se va a ver en la placa.

El día de la evaluación el grupo deberá presentarse 10 minutos antes de la hora establecida en el laboratorio de software del instituto de Ingeniería Eléctrica. Además se deberán traer el KIT DE0-LAB y un “pen drive usb” con todos archivos de los proyectos indicados y las simulaciones realizadas, si se desea usar las pc de la facultad. Si utilizaron una laptop para hacer la tarea se recomienda traer la misma laptop para la defensa. En ambos casos probar que todos los programas funcionan antes de la práctica. Los proyectos deben tener el chip y 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. De utilidad: Para “pegar” en el informe un circuito realizado en el editor gráfico de Quartus seleccionar el circuito, “copiarlo” y realizar un “pegado especial”, indicando “mapa de bits independiente del dispositivo”.