

## PRÁCTICA 3 DESCRIPCIÓN DE HARDWARE (RTL)

### Objetivos

- Realizar una aplicación que utilice un teclado PS2 real utilizando descripción de Hardware RTL.

**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. Las señales CSL mencionadas a lo largo de esta práctica corresponden con las señales PASO vistas en el teórico.

### Introducción

Inicialmente será necesario implementar un receptor de teclado PS2 a partir de los circuitos diseñados en las prácticas 1 y 2.

Como aplicativo del receptor de teclado PS2, se va a diseñar un circuito digital que simule el funcionamiento del display de una canalera de 999 canales, con las operaciones de incrementar canal, ir al canal anterior e ingresar el número de canal marcando en un teclado numérico.

El siguiente esquema representa el simulador para el display de canalera que se quiere implementar.

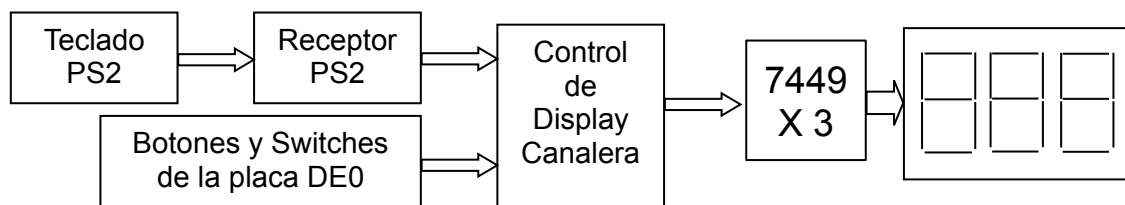


Figura 1. Esquema del simulador de canalera a implementar

El sistema debe funcionar de la siguiente forma:

Inicialmente, luego de un reset, el canal a desplegar deberá ser 555.

#### Ingreso de un nuevo canal

Para ingresar un nuevo canal, al digitar un número  $N_0$  en el teclado numérico PS2, se debe desplegar en el display el valor  $00N_0$ . Si antes de transcurridos 3.5 segundos se digita otro número  $N_1$ , en el display se debe desplegar el valor  $0N_0N_1$ . Y por último, si antes de transcurrir 3.5 segundos **desde que se digitó el segundo número**, se digita un tercer número  $N_2$ , el display debe desplegar el valor  $N_0N_1N_2$ , finalizando el proceso para ingresar un nuevo canal. Este proceso también debe finalizar si en alguna de las primeras 2 ocasiones se superan los 3.5 segundos, debiendo quedar desplegado en el display  $00N_0$  en el primer caso o  $0N_0N_1$  en el segundo.

#### Incremento del canal actual

El canal actual se deberá incrementar mediante el botón **Button0**. Se debe realizar el incremento al momento de soltar el botón.

Cuando se está procesando el ingreso de canal por teclado, se debe hacer caso omiso al incremento y viceversa. En caso de simultaneidad, tiene prioridad el ingreso de un nuevo canal.

Para llegar al sistema completo, se deberá diseñar el circuito detallado en la figura 2, donde:

- Key[3..0] Indica en BCD cual es el valor del número oprimido en el teclado. Es válido sólo cuando *Key\_valid* es 1.
- Key\_valid Sube a 1 durante **un período de reloj** para indicar que *Key[]* es válido.
- Up Al subir esta señal, la salida *CH* deberá incrementarse en 1.
- CH[11..0] Indica en 3 dígitos BCD el canal actual. Debe mantenerse constante hasta el siguiente cambio.

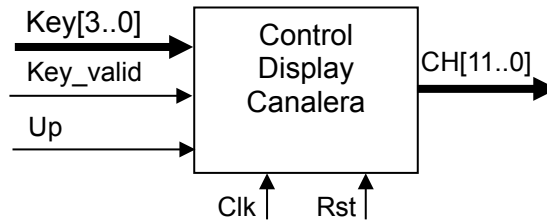


Figura 2. Circuito a diseñar

**Parte a:**

El objetivo de esta parte es solo implementar el bloque **Receptor\_PS2** de la figura 3 y probar su funcionamiento con un teclado PS2 real.

Los bloques **PS2\_BCD**, **SaP\_PS2** y **KEY\_VALID** son los diseñados en las prácticas anteriores.

Observar que el **Receptor\_PS2** captura el código PS2 cuando se suelta cualquier tecla y convierte este código de serie a paralelo. Si la tecla oprimida corresponde al teclado numérico, **Kv=1** por 1 período de reloj, momento en el cual **Num[3..0]** tiene el valor numérico en BCD de dicha tecla.

Se pide:

- Estudiar el funcionamiento del circuito **Receptor\_PS2** de la figura 3.
- Generar un proyecto con el circuito **Receptor\_PS2** y crear el bloque correspondiente.
- Verificar el funcionamiento del **Receptor\_PS2**, probando el circuito de la figura 4 en la placa. Para ello se debe conectar un teclado en el conector PS2 de la placa. Observar que al soltar una tecla numérica, se despliega su valor en el display HEX0\_D.

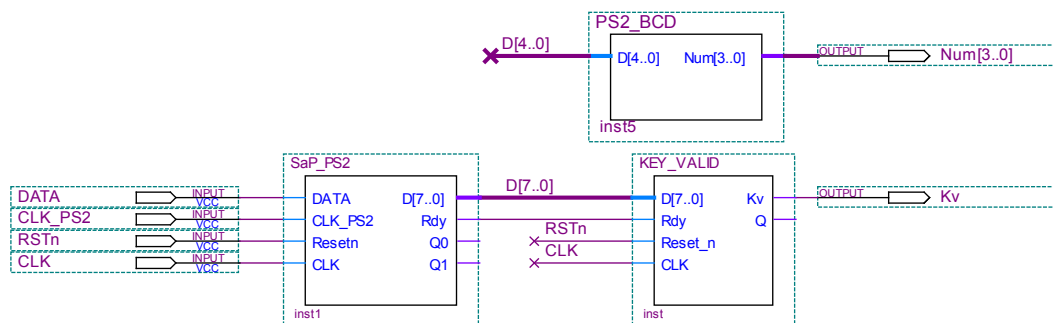


Figura 3. Diagrama del Receptor\_PS2

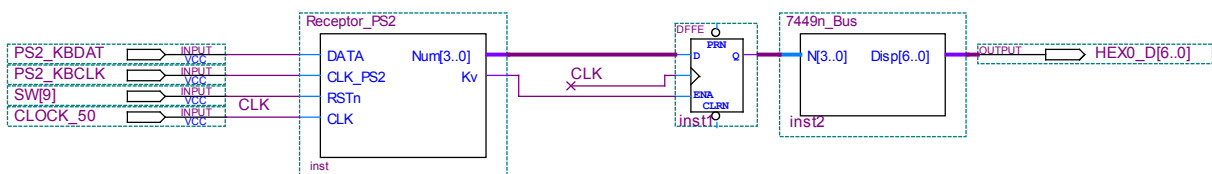


Figura 4. Circuito de prueba de la parte a)

Nota:

- Si en las pruebas se oprimen las teclas de “Inicio”, “←”, “Fin”, “Insert”, etc, el funcionamiento no será el esperado. Esto es debido a que el funcionamiento de un teclado PS2 es mucho más complejo que lo aquí expuesto.
- Si no se dispone de un teclado PS2 utilizar el simulador proporcionado, cuyo instructivo se encuentra en el Anexo.

**Parte b:**

En esta parte se verá un circuito que implementa el ingreso de un canal entre 0 y 99 por teclado y el incremento del mismo. El canal inicial será el 55.

La siguiente descripción RTL, corresponde al circuito **RTL\_A**, que comanda un circuito **Timer** el cual se proporciona. Este **Timer** incrementa su salida Time[2..0] cada 0.5 segundos a partir de que su entrada T es 1. Si T = 0 su salida Time[2..0] = 000. El circuito **INC\_BCD\_2** es el diseñado en la practica 1.

```

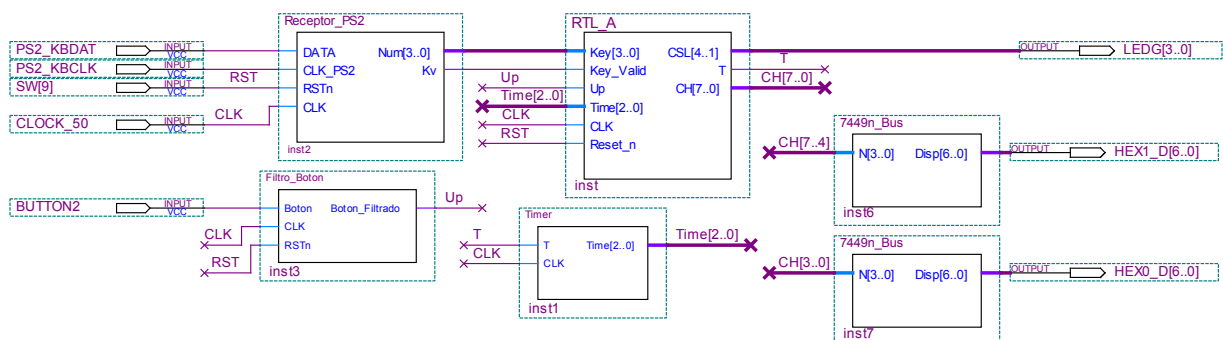
MODULE:      RTL_A
MEMORY:      CH_MEM[7..0]
INPUTS:      Key[3..0], Key_valid, Up, Time[2..0]
OUTPUTS:     CH[7..0], T

1. CH_MEM[] ← 01010101
2. CH_MEM[] * Key_valid ← 0000, Key[3..0]
   →(Key_valid, !Key_valid . !Up, !Key_valid . Up)/(4, 3, 2)
3. CH_MEM[] * Up ← INC_BCD_2(CH_MEM[])
   → (Up, !Up)/(2, 3)
4. T = 1 ;Con T = 1, se comienza a contabilizar 3.5 segundos.
   CH_MEM[] * Key_valid ← CH_MEM[3..0],key[3..0]
   →(Time=111 or Key_valid, !Time=111 and !Key_valid)/(2,4)

END SEQUENCE
CONTROL RESET (1)
    CH[] = CH_MEM[]
END
    
```

Se pide:

- Analizar la descripción RTL dada y la función del circuito **Timer**.
- 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 bloque **RTL\_A\_datos** y **RTL\_A\_control**.
- Generar un proyecto **RTL\_A** a partir de los bloques definidos anteriormente. Se debe agregar CSL[4..1] como salida (ver figura 5) a los efectos de facilitar las pruebas. Simular.
- Crear el bloque **RTL\_A**.
- Generar un proyecto con el circuito de la figura 5. Analizar lo que se deberá observar y el por qué, programar la placa y verificar el funcionamiento.



**Figura 5. Circuito de prueba de la parte b)**

Notas:

- Para evitar los rebotes en el botón, se debe utilizar el circuito proporcionado **Filtro\_Boton**, de acuerdo al circuito de la figura 5.
- El circuito **7449n\_Bus** es el proporcionado en la práctica 2.

**Parte c:**

En esta parte se debe extender la descripción anterior al manejo de los canales entre 0 y 999, tal como se describe en la introducción.

Esto requiere pasar los registros y señales que correspondan de 8 a 12 bits.

Se detalla a continuación como realizar las modificaciones al circuito:

Ingreso de un nuevo canal

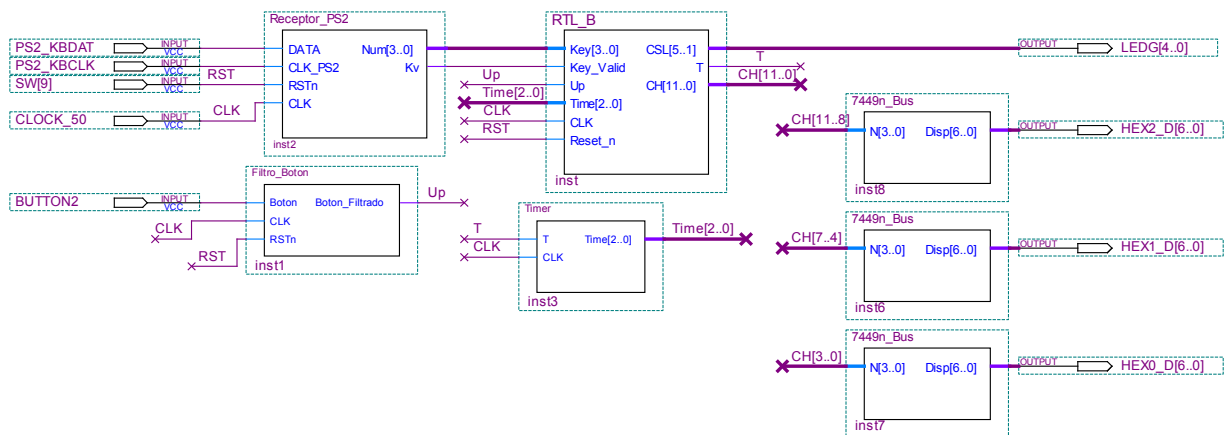
Para el ingreso por teclado se sugiere agregar un paso más (paso 5) en la descripción, de tal forma que en el paso 4 se espera el segundo dígito y en el paso 5 el tercer dígito, dependiendo de los 3.5 segundos en ambos casos. Tener presente que, en el nuevo paso 4, al recibir el segundo dígito o lo que es lo mismo, si Key\_valid es 1, se debe reiniciar la cuenta de tiempo en el Timer.

Incremento del canal actual

Tener presente que para el incremento, no se dispone de un bloque incrementador BCD de 3 dígitos. Se debe realizar este incremento con lenguaje RTL, aplicando una lógica similar a la vista en la práctica 1. Puede ser conveniente tratar a CH\_MEM[7..0] como un registro y a CH\_MEM[11..8] como otro registro. Dado que CH\_MEM[11..8] es un registro de 4 bits, para incrementarlo se debe utilizar el bloque **INC\_BCD\_1** diseñado en la práctica 1.

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 el bloque **RTL\_B** a partir de los bloques definidos anteriormente. Simular.
- Crear el bloque **RTL\_B**.
- Generar un proyecto con el circuito de la figura 6 que es similar al de la figura 5; se cambia **RTL\_A** por **RTL\_B**, se agrega un dígito al display y un led más. Analizar lo que se deberá observar y el por qué, programar la placa y verificar el funcionamiento.



**Figura 6.** Circuito de prueba de la parte c)

## 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)**

- No es necesario incluir esta parte en el informe.

### **Parte b)**

- Diagrama de los circuitos **RTL\_B\_Control**, **RTL\_B\_Datos** y **RTL\_B**.
- Simulaciones.
- Explicar donde en la descripción, se le da prioridad al ingreso por teclado respecto del incremento. ¿Cómo se invertiría la prioridad?

### **Parte c)**

- Descripción **RTL\_C**.
- Breve explicación de los cambios introducidos.
- Diagrama de los circuitos **RTL\_C\_Control**, **RTL\_C\_Datos** y **RTL\_C**.
- Simulaciones.

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”.