

Apellidos, Nombre: _____

Circuito Secuencial Síncrono

*Circuitos Electrónicos Digitales
Ingeniería Informática. Tecnologías Informáticas
Dpto. de Tecnología Electrónica
Diciembre, 2024*

1 Descripción y objetivos

En esta práctica tiene como principal objetivo el diseño de circuitos secuenciales síncronos.

Se trabajan los siguientes conceptos y competencias con respecto a los mismos

- Diseño y análisis
- Implementación mediante elementos discretos.
- Descripción Verilog y simulación
- Uso de Verilog para implementar un circuito haciendo uso de un dispositivo programable.

2 Material y documentación

2.1 Software y hardware

- Instrumentación del laboratorio de Electrónica Digital.
- Ordenador con entorno Xilinx ISE instalado.
- Placa de desarrollo Digilent Basys2.
- Archivos iniciales de diseño (kit de laboratorio).

2.2 Documentación

- Vídeos explicativos.
- Manual del instrumental básico de laboratorio de la asignatura (<https://www.dte.us.es/docencia/etsii/gii-ti/cedti/laboratorio>).
- Tema 7 de la asignatura (https://www.dte.us.es/docencia/etsii/gii-ti/cedti/temario_2012_2013).

3 Primera parte: Estudio teórico

En este apartado vamos a analizar los circuitos con que trabajaremos en el estudio experimental. Serán una implementación de un contador de dos bits ascendente que es también un divisor de frecuencia y un generador de secuencia basado en dicho contador.

En la figura 1 se muestra el contador.

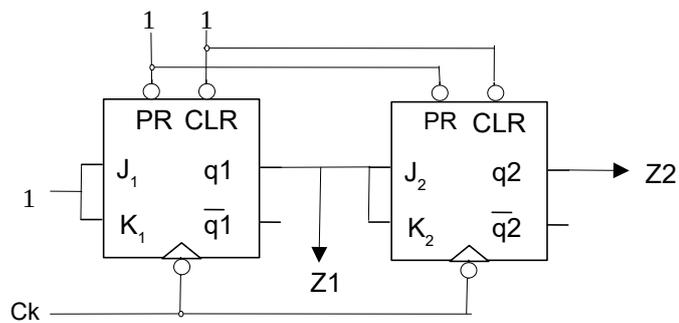
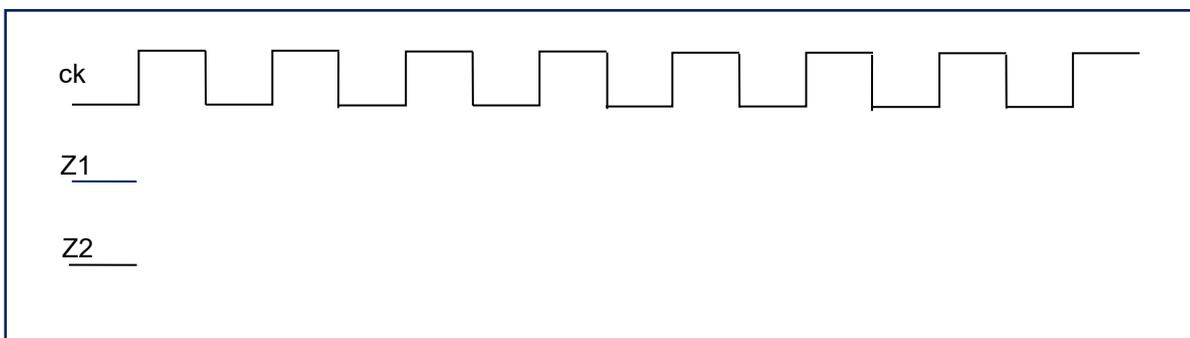


Figura 1

3.1 Obtenga las ecuaciones de excitación salida del circuito.

3.2 Haciendo uso de dichas ecuaciones obtenga el diagrama temporal del circuito. Suponga estado inicial 0 para los dos biestables.



- ¿Qué relación existe entre las frecuencias de ck, Z2 y Z1?
- Escriba la secuencia de valores tomada por Z2 y Z1 en los sucesivos ciclos de reloj:
Z2Z1: 00 → 01 →

En la figura 2 se muestra el generador de secuencia.

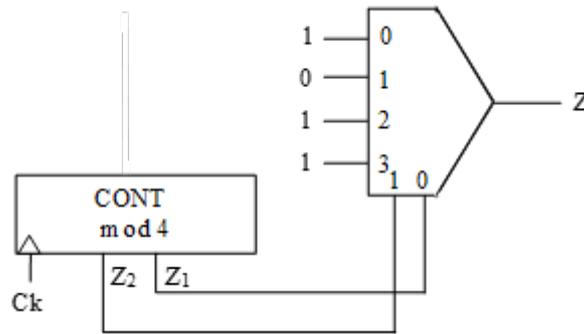
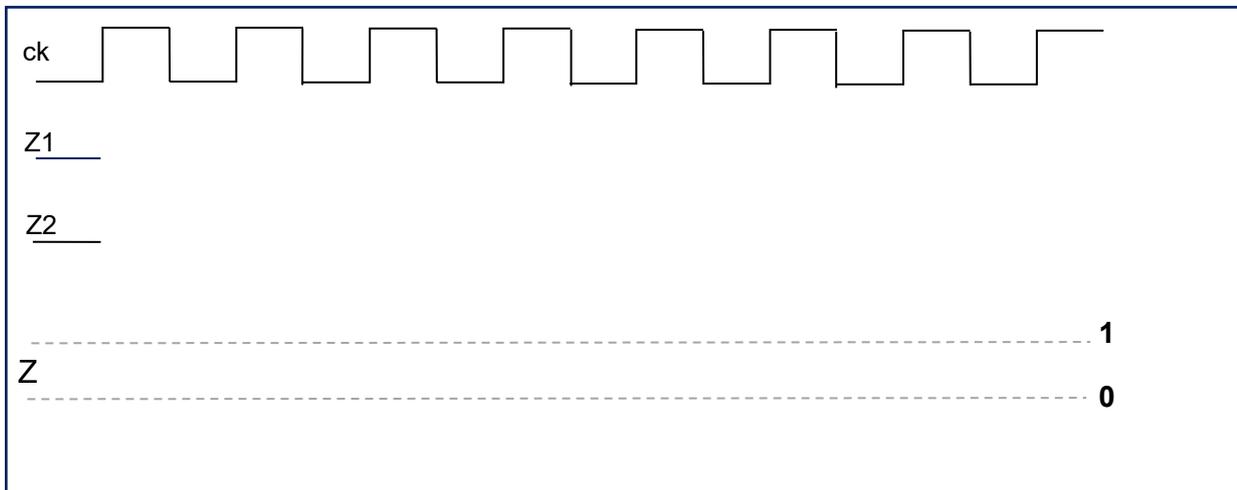


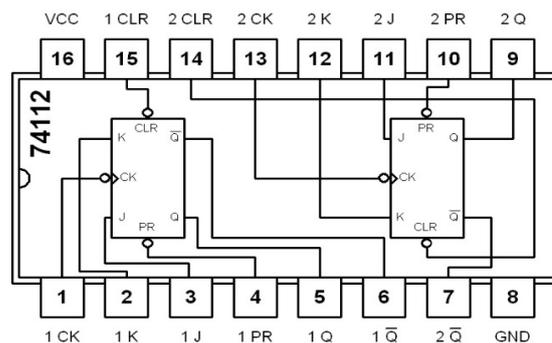
figura 2

3.3 Obtenga el diagrama temporal del circuito. (Estado inicial 00 para el contador)



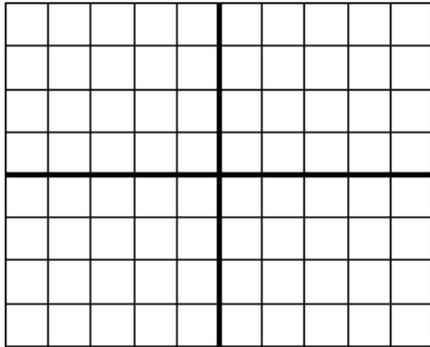
4 Segunda parte: Montaje discreto

4.1 Realice un esquema sobre la figura 3, que indique cómo va a montar el circuito descrito en el estudio teórico (figura 1). Se pide que indique claramente dónde va a colocar entradas y salidas y cómo conectará los circuitos. No olvide fijar a 5V y 0V las entradas VCC y GND de ambos circuitos.



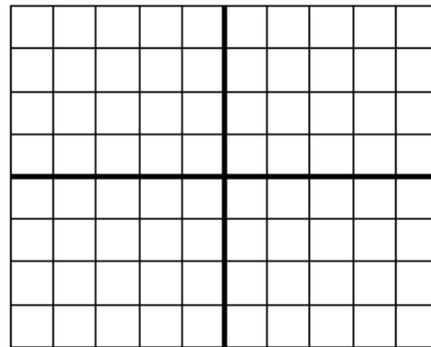
4.2 Compruebe que el montaje funciona correctamente. Para ello, visualice simultáneamente la señal de reloj (Ck) y Z1, y posteriormente las señales Z1 y Z2. Fije como frecuencia de Ck 10Khz. Avise a su profesor para mostrarle los resultados.

4.3. Complete las plantillas para los dos casos anteriores: Ck con Z1 y Z1 con Z2. No olvide señalar la línea de tierra.



Escala de tiempo: _____

Escala de tensión: _____

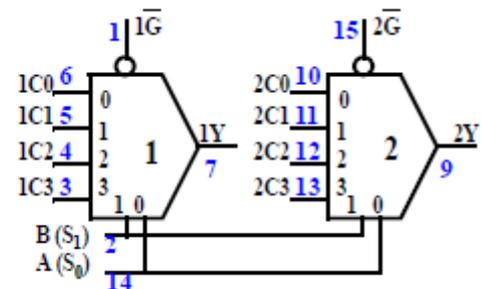
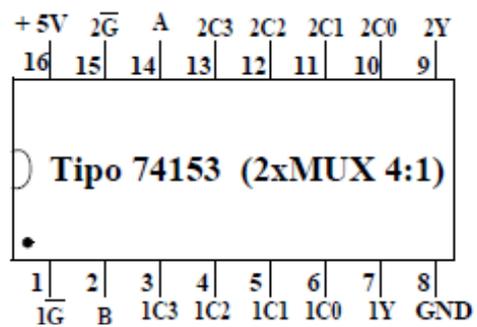
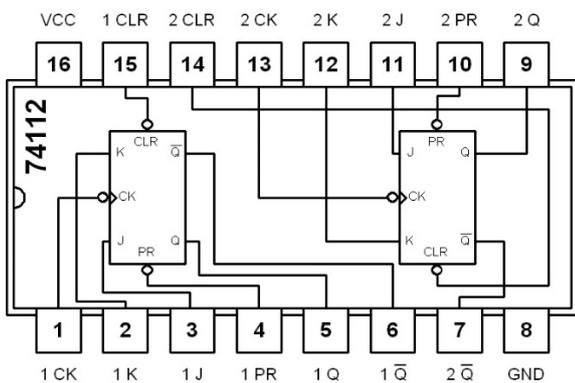


Escala de tiempo: _____

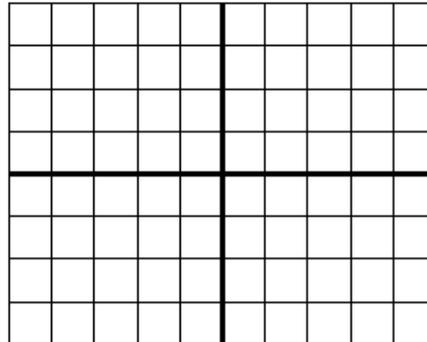
Escala de tensión: _____

☞ **Avise al profesor para mostrarle el funcionamiento**

4.4 Implemente el generador de secuencia (figura 2). Muestre sobre la figura 4 las conexiones que realizará en su montaje.



4.5 Para comprobar su funcionamiento visualice en el osciloscopio las señales Ck y Z simultáneamente y muestre los resultados a su profesor. Refléjelos también en la plantilla .



Escala de tiempo: _____

Escala de tensión: _____

 **Avise al profesor para mostrarle el funcionamiento**

5 Tercera parte: Simulación

Actividades a realizar:

1. Estudie la descripción Verilog del módulo descrito en el archivo `generador.v` que se proporciona. Corresponde a una máquina de Moore y se trata de un generador de secuencia.
2. Dibuje su diagrama de estado.
3. Explique cuál es la secuencia generada.
4. Utilice el testbench para simular el comportamiento del circuito descrito y explique el resultado obtenido.

 **Avise al profesor para mostrarle el funcionamiento**

6 Cuarta parte: Implementación del circuito en FPGA

Para realizar esta parte se trabajará con las descripciones Verilog ya testeadas del circuito en el apartado de simulación de esta práctica.

6.1 Creación de un proyecto en Xilinx ISE

- Inicie el entorno Xilinx ISE.
- Cree un nuevo proyecto desde File → New Project. Se abrirá el asistente para creación de proyectos.
- Ahora introduzca los detalles del tipo de chip FPGA en que se sintetizará el proyecto. Para la placa Basys2 los detalles son:
 - General Purpose
 - Family: Spartan3E
 - Device: XC3S100E
 - Package: CP132
 - Speed grade: -5
 - El resto de opciones del proyecto no deben modificarse: XST synthesis, ISIM simulator, Verilog preferred language, etc.). Pulse NEXT.
 - Revise el resumen de opciones del proyecto y si no hay errores pulse FINISH.

6.2 Añadir archivos al proyecto

- Para añadir los archivos que se le proporcionan, es decir, `generador.v`, `generador_tb.v` y `Basys2_100_250Simplificado.ucf` al proyecto haga click derecho sobre el panel de la jerarquía del diseño (Hierarchy) y elija “Add Source”.
- Seleccione los archivos Verilog. Puede elegir varios archivos a la vez con Ctrl+click. ISE usará estos archivos para implementar y/o simular el diseño. Puede elegir la finalidad de cada archivo manualmente, pero normalmente ISE detectará la función de cada archivo automáticamente y sólo tendremos que confirmar.
- Ahora, el panel de diseño muestra los archivos que forman el proyecto. Hay dos vistas: implementación (Implementation) y simulación (Simulation). La vista de implementación muestra los archivos con el diseño que se configurará en la FPGA. La vista de simulación muestra, además, los archivos con los bancos de prueba que sólo se emplean en la simulación.
- Haciendo doble click sobre cualquier archivo, éste se abre en el panel central y puede ser editado.

6.3 Añadir el fichero UCF para poder realizar la implementación

El chip de la FPGA de la placa Basys2 tiene sus pines conectados a varios periféricos: pulsadores, interruptores, LEDs, display 7 segmentos, ect. Para poder implementar el diseño es necesario mapear las entradas y salidas de nuestro diseño a los pines adecuados del dispositivo FPGA de manera que queden en periféricos útiles de la placa. Un mapeado correcto enlazaría entradas con interruptores o pulsadores y salidas con LEDs. El fichero UCF contiene la información de este mapeado y es necesario incluirlo en el proyecto y editarlo de acuerdo a las necesidades de nuestro diseño y la placa con la FPGA que se esté utilizando.

Para este circuito utilizaremos **uno de los switches como entrada x (p. ej. sw<0>)** y conectaremos **la salida z a uno de los led (p. ej. Led<0>)**, también conectaremos **la señal clk a un botón de la placa (p. ej. BTN3)** para generar el reloj mediante pulsaciones. La placa tiene entradas de reloj que en principio podrían utilizarse, pero la frecuencia de operación de este reloj es muy alta y eso hará que no se distinga el apagado y encendido de los leds, por esta razón generaremos un reloj haciendo uso de uno de los pulsadores. En general, las señales de reloj deben suministrarse al sistema por pines (IOB) especiales para señales de reloj, pero nosotros hemos conectado esta señal a un botón de la placa que está asignado a un IOB convencional. Como esto es en general un error de diseño, la herramienta se detiene en la fase de implementación. Nuestro diseño es muy simple y no hay problema en tener el reloj en un pin convencional. Podemos indicar a la herramienta que ignore esta restricción en el archivo UCF. Edite el archivo UCF y añada una regla que indique que la señal “clk” no tiene que usar una ruta de reloj especial. Para ello **copie la siguiente línea** tras la línea de asignación de la señal “clk”:

```
NET "clk" LOC = "A7"; # Bank = 1, Signal name = BTN3 +
NET "clk" CLOCK_DEDICATED_ROUTE = FALSE;
```

6.4 Síntesis e Implementación

- Seleccione el módulo a implementar en el panel de diseño. En el panel de procesos aparecen las tres acciones principales del proceso completo de síntesis: *Synthesize - XST, Implement Design* y *Generate Programming File*.
- Haga click derecho sobre *Generate Programming File* y seleccione *Run*. Esto generará el archivo de configuración de la FPGA, pero también ejecutará todos los procesos anteriores que sean necesarios. Observe el icono animado que indica la ejecución de cada proceso.

Si todo va bien, aparecerá una marca verde o amarilla junto a cada proceso. Si algún proceso falla, aparecerá una marca roja y se detendrá la síntesis. En este caso, puede consultar los errores en la consola inferior, en la pestaña *Errors*. Mire si hay errores, lea los mensajes de error e intente interpretarlos

6.4.1 Programación del diseño en la FPGA

Como resultado de los procesos anteriores, se ha generado un archivo con la configuración (bitstream) que hay que cargar en el dispositivo FPGA. Esta configuración está en el archivo `.bit` guardado en la carpeta del proyecto. El proceso de programación consiste simplemente en cargar este paquete de bits en la FPGA.

 **Avisé al profesor para mostrarle el funcionamiento del circuito.**

7 Cuarta parte: proyectos adicionales

En esta parte el alumno deberá realizar la simulación e implementación en la FPGA de un circuito secuencial síncrono que genere una secuencia distinta a la propuesta.