

Apellidos, Nombre: _____

Diseño de Circuito Combinacional mediante SSI y con Dispositivos programables

*Circuitos Electrónicos Digitales
Ingeniería Informática. Tecnologías Informáticas
Dpto. de Tecnología Electrónica
Septiembre, 2022*

1 Descripción y objetivos

En esta práctica se pretende abordar el diseño de un circuito combinacional mediante dos procedimientos:

- El uso de Circuitos integrados de escala de integración pequeña (SSI- Small Scale Integration).
- Mediante dispositivos programables.

Se trabajan los siguientes conceptos y competencias:

- Diseño de Circuitos Combinacionales.
- Implementación mediante elementos discretos de Circuitos Combinacionales.
- Uso del instrumental habitual de laboratorio de electrónica.
- Uso de lenguajes de descripción de hardware (en concreto Verilog) para simulación de circuitos combinacionales.
- Uso de Verilog para implementar un circuito combinacional haciendo uso de un dispositivo programable.

La práctica se estructura a partir de un proyecto inicial que es explicado íntegramente por el profesor a modo de ejemplo (bien en directo o mediante un vídeo pregrabado). A partir de este proyecto de ejemplo, se proponen varios proyectos adicionales que amplían o se basan en el proyecto inicial. El alumno debe reproducir el proyecto inicial y realizar uno o varios de los proyectos adicionales.

2 Material y documentación

2.1 Software y hardware

- Herramienta web de simulación de circuitos a partir de su descripción en Verilog.
- Instrumentación del laboratorio de Electrónica Digital.
- Circuitos integrados 7400 y 7410.
- Software para simulación de circuitos electrónicos a a partir de su descripción en Verilog (EDA Playground o Icarus Verilog)
- Ordenador con entorno Xilinx ISE instalado.
- Placa de desarrollo Digilent Basys2.
- Archivos iniciales de diseño (kit de laboratorio).

2.2 Documentación

- Vídeos explicativos en Enseñanza Virtual
- Manual del instrumental básico de laboratorio de la asignatura en https://www.dte.us.es/docencia/etsii/gii-ti/cedti/laboratorio_2_4
- Temario de la asignatura
- Diagrama de pines de los CI 7400 y 7410. Documento patillaje en https://www.dte.us.es/docencia/etsii/gii-ti/cedti/laboratorio_2_4/patillajes/view

3 Primera parte: Simulación


Actividades a realizar:

1. Visualice los videos explicativos sobre la instalación del software de simulación a usar en esta práctica.
2. Visualice el video 1 sobre la descripción en Verilog del circuito votador
3. Complete los ficheros de diseño proporcionados para describir el circuito votador mediante las tres formas posibles: funcional, estructural y procedimental.
4. Visualice los videos 2 y 3 sobre la descripción del fichero de testbench y el uso de los simuladores para circuitos en Verilog.
5. Con el testbench proporcionado simule las tres descripciones del votador utilizando el simulador de su elección. Si es necesario, corrija las descripciones hasta que el comportamiento sea correcto.

3.1 Trabajo a entregar por el alumno correspondiente a la primera parte

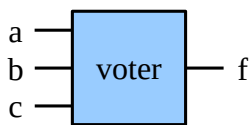
Se deberá entregar un fichero que contendrá:

- Ficheros fuente de las descripciones Verilog del votador completados y correctos
- Cronogramas generados en las simulaciones

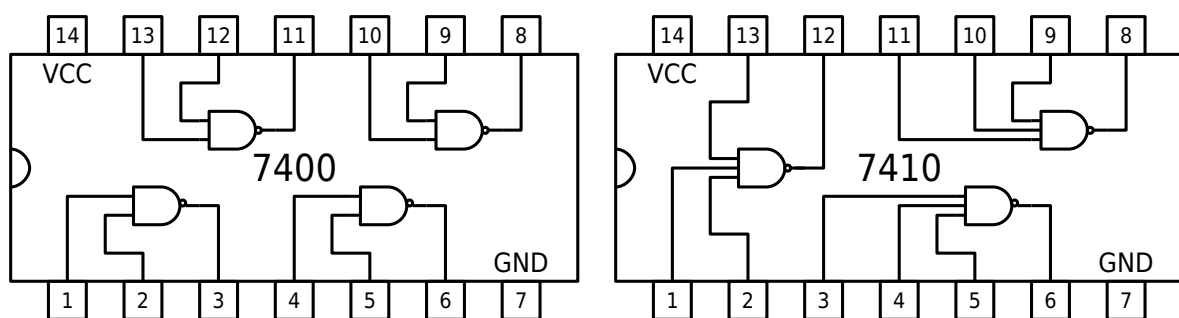
 **La realización de la práctica requiere entregar previamente este fichero dentro del plazo establecido.**

4 Segunda Parte: Montaje del circuito votador mediante CI SSI

1. Diseñe un circuito votador con tres entradas (a,b,c) y una salida usando solo puertas NAND.



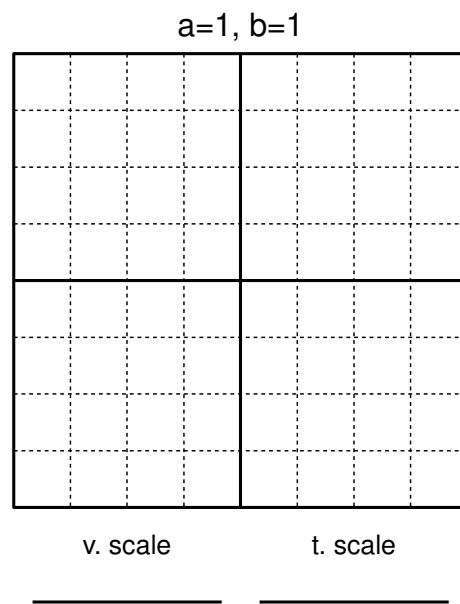
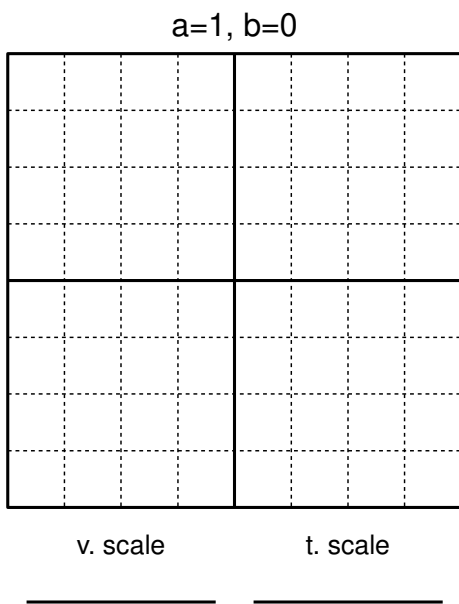
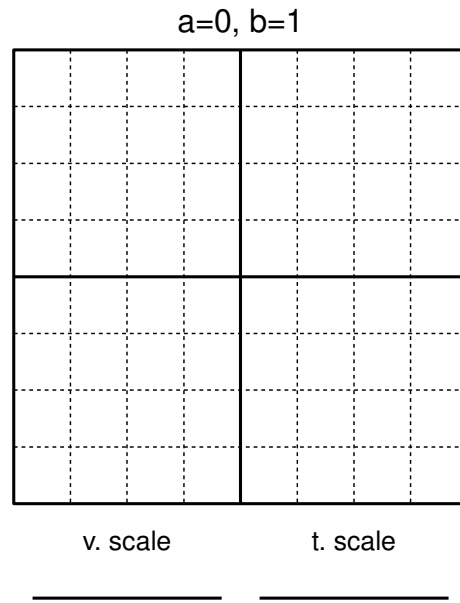
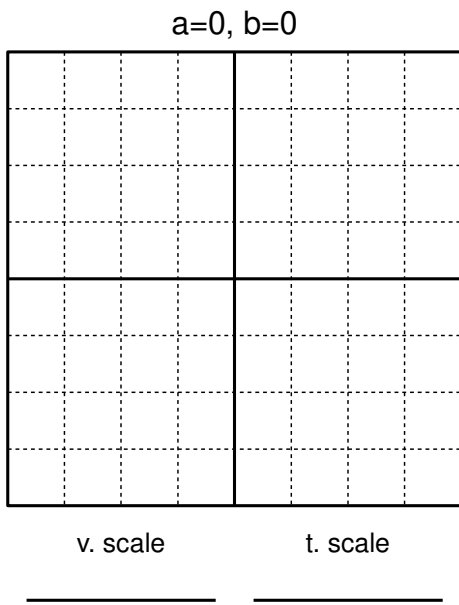
2. Dibuje las interconexiones utilizando CI 7400 y 7410.



3. Monte el circuito utilizando la regleta de ensayo. Alimente los CIs
4. Genere una señal cuadrada de 50Khz de 0 a 5 V. Conecte la señal cuadrada a la entrada c.

Realice un test al circuito para las condiciones que se detallan a continuación dibujando las formas de onda c y f observadas en el osciloscopio.

- a) $a=0, b=0$ b) $a=0, b=1$ c) $a=1, b=0$ d) $a=1, b=1$




☞ Avise al profesor para que de el visto bueno al montaje una vez que hay comprobado su correcto funcionamiento.

5. Deduzca la tabla de verdad basada en los resultados medidos.

6. Para el caso $a=0$ y $b=1$, mida el retardo de propagación entre c y f tanto en subida como en bajada (t_{pHL} y t_{pLH}).

$t_{pHL} =$

$t_{pLH} =$

 **Avisé al profesor cuando tenga la configuración correcta para la medida del tiempo de propagación.**

5 Tercera parte: Implementación del votador en FPGA

Para realizar esta parte escoja una de las tres formas de descripción del circuito votador en Verilog.

5.1 Creación de un proyecto en Xilinx ISE

- Inicie el entorno Xilinx ISE. La ventana del entorno se divide en tres secciones: la zona izquierda es el panel del proyecto y sirve para gestionar los archivos y módulos del proyecto y las acciones que se pueden realizar sobre los mismos; la zona derecha es el panel principal que da acceso a la edición de los elementos del proyecto, como los archivos Verilog; la zona inferior es la consola, donde podemos ver los mensajes de error y avisos de las herramientas del entorno conforme se realizan las operaciones sobre el diseño.
- Es posible que al iniciar ISE se abra automáticamente el último proyecto editado. Si es así, cierre el proyecto desde File → Close Project.
- Cree un nuevo proyecto desde File → New Project. Se abrirá el asistente para creación de proyectos. Escriba un nombre para el proyecto como “voter”. Puede elegir la carpeta para el

nuevo proyecto si lo desea, por ejemplo, la carpeta donde ha descargado los archivos iniciales de la práctica. Pulse NEXT.

- 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.

5.2 Añadir archivos al proyecto

- Para añadir los archivos previamente elaborados al proyecto haga click derecho sobre el panel de la jerarquía del diseño (Hierarchy) y elija “Add Source”.
- Seleccione los archivos Verilog “voter.v” y “voter_tb.v”. 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.

5.3 Simulación del banco de pruebas

Aunque en la primera parte de esta práctica ya se ha realizado una simulación del circuito, no está demás realizar una simulación en el entorno ISE previa a la implementación.

Para simular el módulo “voter” con el banco de pruebas suministrado haga lo siguiente:

Seleccione la vista de simulación y seleccione el módulo “test” que se encuentra en el archivo “voter_tb.v”.

- Observe como el panel jerárquico organiza el diseño en función de los módulos que contiene, independientemente de en qué archivo se definen (aunque indica el nombre del archivo entre paréntesis).
- En el panel de procesos (Processes) aparecen las acciones que se pueden ejecutar sobre el módulo seleccionado en el panel de vistas del diseño. Al seleccionar el módulo “test”

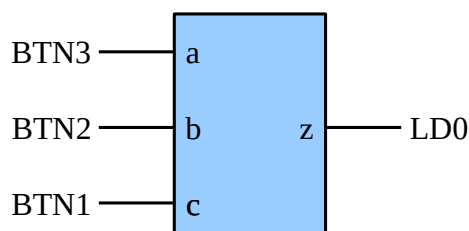
aparecen las acciones asociadas al simulador ISim. Aquí puede seleccionar comprobar la sintaxis del código Verilog (Behavioral Check Syntax) o directamente simular el módulo del banco de pruebas (Simulate Behavioral Model). Haga doble click sobre esta última opción. Si hay errores en el diseño tendrá que editar el código y volver a ejecutar la simulación.

- Si el código es correcto, tras unos segundos se abrirá la ventana del simulador ISim. En el panel principal verá el código Verilog simulado y una marca donde se ha detenido el simulador. Elija la pestaña con las formas de onda, busque los botones de selección de ampliación (zoom) y use el botón que muestra todo el tiempo simulado. Observe las señales y compruebe que el resultado es correcto.
- Por defecto, ISim simula un tiempo de 1000ns o bien hasta que encuentre la directiva Verilog \$finish. Desde el menú “Simulation” o mediante los botones de control de la simulación sobre el panel principal puede continuar la simulación, detenerla o reiniciarla, pero no es necesario para este ejercicio.
- El formato en que se muestran las señales puede modificarse desde el menú contextual que aparece al hacer click derecho sobre cualquier señal. Por ejemplo, pruebe a mostrar la señal de 4 bits “q” en formato decimal sin signo: Radix → Unsigned Decimal.
- Es muy útil guardar la configuración de la visualización de las señales para futuras simulaciones. Elija en el menú File → Save As... y guarde la configuración con un nombre significativo como “voter_tb.wcfg”.
- Salga del simulador. De vuelta en la ventana de ISE, vamos a configurar el proceso de simulación para que cargue automáticamente la configuración de ondas que hemos salvado. Para ello, haga click derecho sobre el proceso “Simulate Behavioral Model” y elija “Process Properties...”. En la ventana que aparece, busque la opción “Custom Waveform Configuration File” y busque el archivo de configuración de ondas “voter_tb.wcfg”. Esto facilitará mucho ver los nuevos resultados de simulación cuando volvamos a ejecutar el simulador.

5.4 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 el circuito votador mapearemos las entradas a,b y c en los tres pulsadores de la placa y la salida z en uno de los LEDs.



- Añada el fichero UCF al proyecto.
- Abra el fichero UCF haciendo doble click en el panel del navegador de la jerarquía de archivos. El fichero suministrado es un fichero genérico para la placa Basys2 que tiene todas las líneas comentadas. Nosotros únicamente quitaremos los comentarios y configuraremos 4 líneas correspondientes a las tres entradas y a la salida del circuito.
 - Primero mapearemos la señal de salida “z” a uno de los LEDs en la placa. Encuentre la línea del fichero que corresponde al LED 0 (signal name LD0):


```
#NET "Led<0>" LOC = "M5" ; # Bank = 2, Signal name = LD0
```
 - Descomente la línea (borrar “#” del principio) y borre el nombre “Led<0>” cambiándolo por el nombre de la señal de salida (“z”):


```
NET "z" LOC = "M5" ; # Bank = 2, Signal name = LD0
```
 - Ahora se trata de mapear los tres pulsadores en las tres entradas. Encuentre las líneas correspondientes a los pulsadores. Ej BTN3:


```
#NET "btn<3>" LOC = "A7"; # Bank = 1, Signal name = BTN3
```
 - Mapee BTN3 a la entrada “a”:


```
NET "a" LOC = "A7"; # Bank = 1, Signal name = BTN3
```
 - De la misma forma , mapee BTN2 a la entrada “b” y BTN1 a la entrada “c”
- Salve el fichero UCF.

5.5 Síntesis e Implementación

“Síntesis” es el proceso que convierte el código Verilog en un circuito digital que realiza la operación descrita por dicho código. Es un proceso complejo similar el proceso manual consistente en obtener la tabla de verdad de las funciones, reducir las funciones (K-mapa), seleccionar las puertas lógicas adecuadas, etc. La “implementación” consiste en varios pasos en los que la funcionalidad del circuito obtenido por la síntesis se “mapea” en los bloques lógicos configurables (CLBs) que componen la FPGA y se establecen las rutas de conexión de estos bloques para obtener un circuito funcional. El circuito generado por la síntesis es un circuito compuesto de componentes estándar (puertas lógicas y biestable) independiente del tipo de FPGA usado. El proceso de implementación depende del chip de FPGA disponible y de los elementos que contenga. Es durante el proceso de implementación que se tienen en cuenta las restricciones para la conexión de señales y pines descrita anteriormente.

El proceso de implementación finaliza generando el conjunto de bits (*bitstream*) de configuración de los CLBs y matrices de interconexión de la FPGA. Cuando estos bits sean cargados en los bloques correspondientes, la FPGA se comportará como el circuito diseñado.

Por suerte, las herramientas son capaces de realizar los procesos de síntesis e implementación de forma completamente automática, aunque es posible modificar parámetros para obtener diferentes resultados: optimizar velocidad, consumo de potencia, etc.

- Seleccione el módulo a implementar en el panel de diseño (módulo **voter**). 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

5.5.1 Implementar el 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 *voter.bit* guardado en la carpeta del proyecto. El proceso de programación consiste simplemente en cargar este paquete de bits en la FPGA. El proceso es sencillo, pero existen varias alternativas para hacerlo, dependiendo del sistema operativo donde estemos usando ISE:

1. Emplear la herramienta de configuración del entorno ISE (iMPACT). Disponible en MS-Windows y GNU/Linux.
2. Emplear la herramienta independiente *Adept* suministrada por el fabricante de la placa Basys2. Disponible en MS-Windows.
3. Emplear el comando de configuración *djtgcfg* suministrado por el fabricante de la placa. Disponible en GNU/Linux.

Con cualquiera de las opciones, tendremos el diseño programado en la placa y podremos probarlo. En caso de que la operación no sea la correcta debemos comprobar todo el proceso de diseño:

- Revise si las herramientas han dado errores o avisos en algún momento del proceso.
- Compruebe que la asignación de señales a pines de la placa son correctos en el archivo UCF.
- Compruebe si los resultados de la simulación son correctos. Si no hemos hecho un banco de pruebas y simulado previamente el diseño, quizá sea hora de hacerlo.

A continuación se describen los procedimientos de configuración de la FPGA. Sólo tiene que usar uno de ellos.

Configuración desde el entorno ISE (iMPACT)

El procedimiento es el siguiente:

- Asegúrese de que la placa está conectada a un puerto USB y tiene accionado el conmutador de encendido.
- Haga click derecho en el proceso *Configure Target Device* y seleccione *Run*. La herramienta iMPACT se abrirá en una nueva ventana.
- En el panel *iMPACT Flows* (arriba a la izquierda) haga doble click en *Boundary scan*. Se creará un nuevo proyecto de programación en el panel principal.
- Haga click derecho en el panel principal y seleccione *Cable Setup...* En la ventana que aparece, seleccione *Open Cable Plugin* y escriba *digilent_plugin* en la entrada de texto. Pulse OK. Con esto hemos configurado la herramienta para que use el complemento de

configuración suministrado por el fabricante de la placa, que ha debido ser instalado en el sistema previamente.

- Haga click derecho en el panel principal y seleccione *Initialize Chain*. La herramienta ahora detectará la placa y analizará qué dispositivos contiene. Cierre las ventanas que aparecen eligiendo “No” y/o “Cancel”.

Ahora puede ver que la placa Basys2 tiene dos dispositivos que pueden ser programados, la propia FPGA (xc3s100e) y una memoria ROM (xcf02s). Para probar nuestro diseño lo programaremos directamente en la FPGA. El problema es que al desconectar la placa la configuración de la FPGA se borrará. Podemos guardar la configuración de forma permanente en el chip EEPROM, pero esta opción no la veremos en esta práctica.

Para configuración la FPGA:

- Haga click derecho en el icono del dispositivo xc3s100e y elija *Assign New Configuration File...*
- Seleccione el archivo `voter.bit` en la carpeta del proyecto y pulse *OK*.
- Vuelva a hacer click derecho en el icono de la FPGA y seleccione *Program*. Uno de los LEDs de la FPGA parpadeará unos segundos. Luego el dispositivo estará configurado.
- Pruebe el diseño.
- Cierre la ventana de iMPACT.

Configuración empleando la herramienta gráfica Adept

Esta herramienta sólo está disponible para sistemas MS-Windows. El procedimiento es el siguiente:

- Asegúrese de que la placa está conectada a un puerto USB y tiene accionado el conmutador de encendido.
- Ejecute la herramienta *Digilent Adept*. Debe aparecer una ventana con el nombre de la placa (Basys2) y se deben mostrar dos entradas para cargar archivos de configuración para los dos dispositivos que tiene la placa: FPGA y PROM.
- En la pestaña *Config* introduzca el nombre del archivo a configurar en la entrada que corresponde al dispositivo FPGA, buscando el archivo en el sistema de archivos.
- Pulse en *Program* junto a la entrada de la FPGA. Es posible que aparezca una ventana de aviso advirtiéndole que el *bitstream* está configurado para usar CCLK en vez de JTAG CLK como reloj de inicio de la FPGA. Puede continuar e ignorar el aviso, y consultar la nota de diseño 1 al final de este manual para una explicación sobre este aviso.
- Uno de los LEDs de la FPGA parpadeará unos segundos. Luego el dispositivo estará configurado.
- Pruebe el diseño.
- Cierre la ventana de *Digilent Adept*.

Configuración empleando el comando `djtgcfg`

El comando `djtgcfg` solo está disponible para sistemas GNU/Linux en los que se ha instalado el software del fabricante de la placa (Digilent).

El procedimiento es el siguiente:

- Asegúrese de que la placa está conectada a un puerto USB y tiene accionado el conmutador de encendido.
- Abra un terminal y cambie la carpeta de trabajo a la carpeta del proyecto.

```
$ cd <carpeta_proyecto>
```

- Liste las tarjetas conectadas al sistema:

```
$ djtgcfg enum
Found 1 device(s)
Device: Basys2
  Product Name: Digilent Basys2-100
  User Name: Basys2
  Serial Number: 210155260217
```

- Inicialice la placa Basys2 y liste los dispositivos disponibles:

```
$ djtgcfg init -d Basys2
Initializing scan chain...
Found Device ID: f5045093
Found Device ID: 11c10093
Found 2 device(s):
  Device 0: XC3S100E
  Device 1: XCF02S
```

- Configure el dispositivo 0 (FPGA) con el archivo de programación `counter.bit`. Es posible que la herramienta de un aviso asociado al reloj de inicio, como en el ejemplo de abajo. Elija “Y” para aceptar la programación a pesar del aviso y consulte la nota de diseño 1 al final de este manual para una explicación sobre este aviso. Uno de los LEDs de la FPGA parpadeará unos segundos. Luego el dispositivo estará configurado.

```
$ djtgcfg prog -d Basys2 -i 0 -f counter.bit
WARNING: startup clock for this file is 'CCLK' instead of 'JTAG
CLK'.
Problems will likely occur. Associate config file with device
anyway (Y or N)? [N] Y
Programming device. Do not touch your board. This may take a few
minutes... Programming succeeded.
```

- Pruebe el diseño

6 Cuarta parte: proyectos adicionales

En esta parte el alumno deberá realizar la simulación e implementación en la FPGA de un circuito combinacional con alguna utilidad práctica.

La simulación será un trabajo previo tal como se describió en la primera parte de esta práctica.

Algunas opciones:

- Se tiene un sensor que proporciona la temperatura mediante valores binarios de 4 bits. Diseñar un circuito que active la refrigeración cuando la temperatura es superior a 12 y la calefacción cuando es inferior a 3.
- Se tienen dos sensores de presencia en un almacén de dos puertas. Diseñar un sistema que genere un zumbido si está activa la alarma y se detecta presencia.
- Proponga su propia aplicación.