

Apellidos, Nombre: \_\_\_\_\_

## **Subsistemas Combinacionales**

*Circuitos Electrónicos Digitales  
Ingeniería Informática. Tecnologías Informáticas  
Dpto. de Tecnología Electrónica  
Noviembre 2023*

### **1 Descripción y objetivos**

---

Esta práctica tiene como principal objetivo el trabajo con circuitos aritméticos. Para ello se usará lenguaje Verilog y dispositivos programables.

Se trabajan los siguientes conceptos y competencias:

- Diseño de circuitos aritméticos.
- Uso de lenguajes de descripción de hardware (en concreto Verilog) para simulación de subsistemas aritméticos.
- Uso de Verilog para implementar un circuito haciendo uso de un dispositivo programable.

La práctica se estructura en dos partes:

- La primera parte (simulación) se debe completar antes de acudir al laboratorio.
- La segunda parte se realizará en el laboratorio.

### **2 Material y documentación**

---

#### **2.1 Software y hardware**

- 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.
- Tema de circuitos aritméticos.

## 3 Primera parte: Simulación

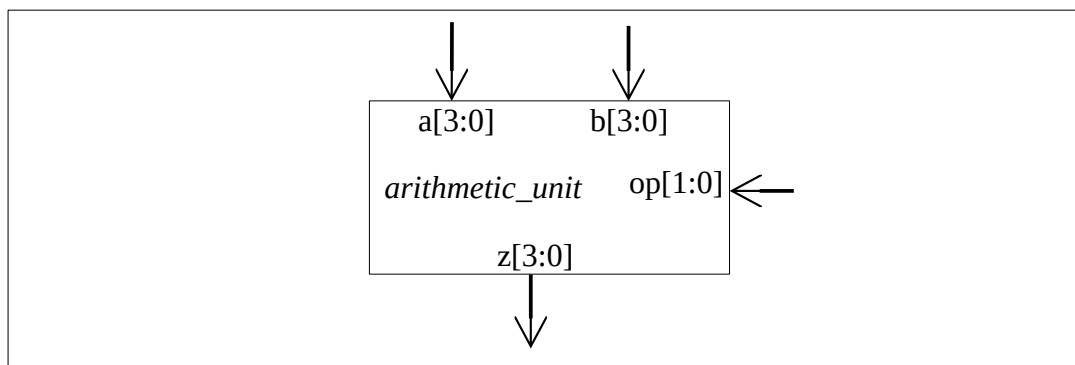
Actividades a realizar (resumen):

1. Realizar la descripción Verilog de una unidad aritmética.
2. Realizar un módulo calculadora a partir de la unidad aritmética y un convertidor 7 segmentos.
3. Utilizar archivos de testbench para simular el comportamiento de los circuitos descritos.

### 3.1 Descripción Verilog de la Unidad Aritmética

En este apartado se describen los pasos a seguir para completar el código y realizar las simulaciones de los circuitos.

Observe el diagrama entrada/salida de la unidad aritmética:



1. Complete la descripción Verilog de la unidad aritmética (“unidad\_aritmetica.v”) cuyas operaciones se muestran en la tabla 2.

Tabla 2. Operaciones de la UA

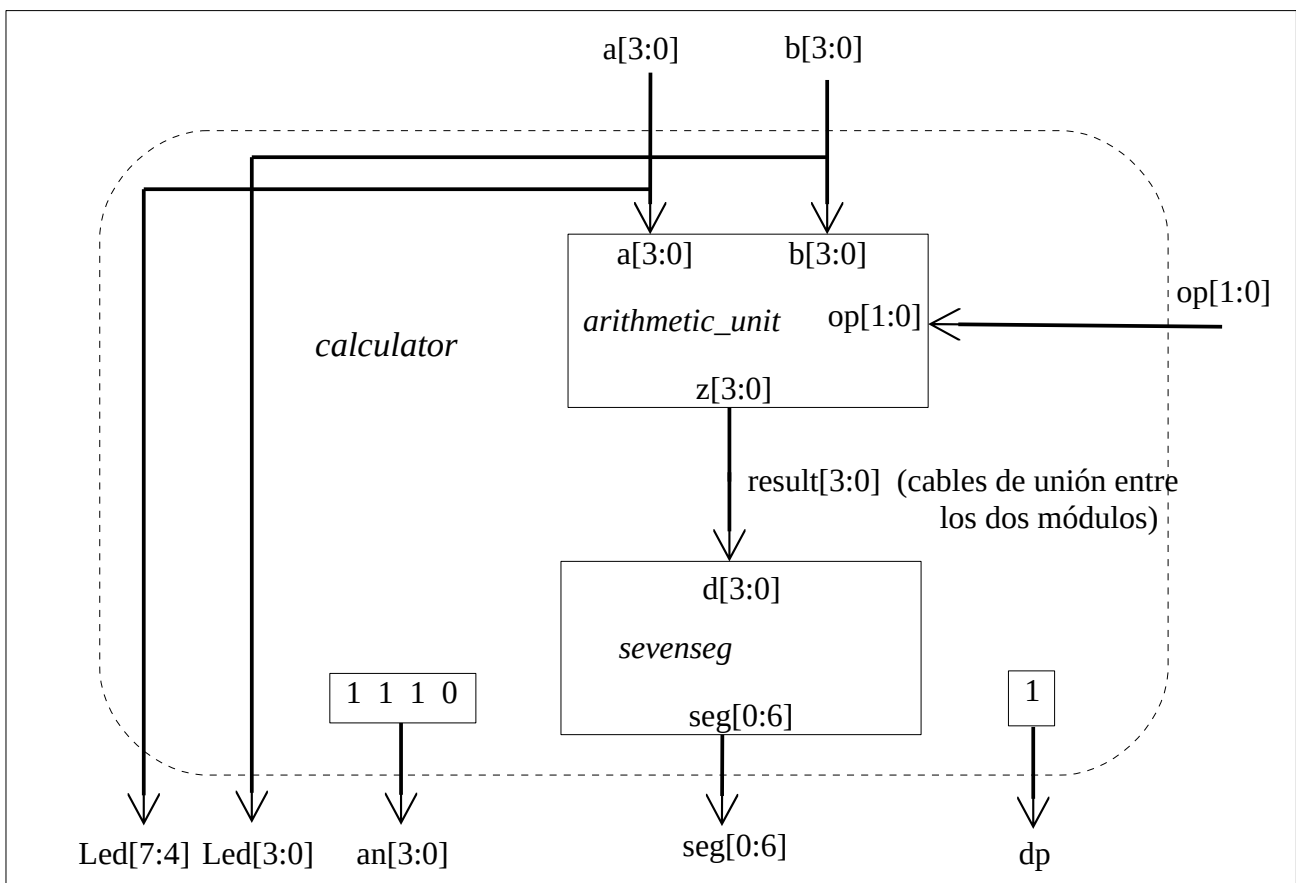
<b>OP<sub>1</sub> OP<sub>0</sub></b>	<b>Z</b>
0 0	a + b
0 1	a – b
1 0	a
1 1	b

2. Compruebe su funcionamiento mediante el testbench (“unidad\_aritmetica\_tb.v”).

Para probar la unidad aritmética en la placa de la FPGA es necesario describir un nuevo módulo (“calculator.v”) en el que se conectará la unidad aritmética con un convertidor binario/7 segmentos (“sevensseg.v”). También se incluyen otros elementos (observe la figura).

Mediante las salidas  $an[3:0]$ ,  $dp$  y  $seg[0:6]$  se controlan los visualizadores 7-segmentos de la placa de la FPGA. En la placa existen 4 visualizadores pero solo se trabajará con uno. Para seleccionar uno entre los 4 se utilizan las salidas  $an[3:0]$  que deben fijarse a los valores 1110. El bit que está a 0 selecciona el visualizador deseado. La salida  $dp$  también se fija a 1 para desactivar el punto decimal del visualizador.

Por otra parte también se conectan las entradas  $a[3:0]$  y  $b[3:0]$  a las salidas  $Led[7:4]$  y  $Led[3:0]$  respectivamente para poder visualizar en los leds de la placa los valores de los datos  $a$  y  $b$ .



3. Revise la descripción Verilog del convertidor 7 segmentos (“sevensseg.v”) que se le proporciona.
4. Complete la descripción Verilog de la calculadora (“calculator.v”). En ella ya aparecen instanciados todos los módulos de la figura excepto la unidad aritmética, por tanto, necesitará instanciar la unidad aritmética y conectarla al módulo *sevensseg* utilizando los cables de unión.

## Trabajo a entregar por el alumno correspondiente a la 1ª parte

Se deberá entregar un documento que contendrá:

- Descripciones Verilog de los circuitos propuestos, para ello incluya los ficheros fuente con el código Verilog y no utilice capturas gráficas del texto.
- Resultados obtenidos en la simulación y explicación de los mismos, en este caso si puede incluir la captura de pantalla de los cronogramas.

## 4 Segunda parte: Implementación de la calculadora en FPGA

---

Para realizar esta parte se trabajará con las descripciones Verilog de la calculadora realizada según las instrucciones del apartado 3.1 de esta práctica.

### 4.1 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 “calculadora”. 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.

## 4.2 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 “calculator.v”, “calculator\_tb.v”, “sevensseg.v”, “unidad\_aritmetica.v” y “unidad\_aritmetica\_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, este se abre en el panel central y puede ser editado.

## 4.3 Simulación del banco de pruebas.

En esta y siguientes secciones deberá utilizar la descripción Verilog de la unidad aritmética (“unidad\_aritmetica.v”) y del módulo calculator (“calculator.v”) que habrá completado según se pide en el apartado 3.1

Para simular el módulo “unidad\_aritmética” 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 “unidad\_aritmetica\_tb.v”.

- Observe cómo 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 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 que indica dónde 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 (1µs) 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 caso concreto.

- El formato en que se muestran las señales puede modificarse desde el menú contextual que aparece al hacer click derecho sobre cualquier señal.

Para simular el módulo “calculator” siga los mismos pasos con el banco de pruebas correspondiente, para ello, en la vista de simulación seleccione el módulo “test2” que se encuentra en el archivo “calculator\_tb.v”.

## 4.4 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.

En este caso, mapearemos las entradas  $a[3:0]$  y  $b[3:0]$  en los conmutadores o switches de la placa ( $sw<7>$ ,  $sw<6>$ , ...,  $sw<0>$ ), las señales de selección de operación en los pulsadores BTN1 y BTN0 y las salidas  $seg[0:6]$  en el visualizador 7 segmentos ( $seg<0>$ ,  $seg<1>$ , ...,  $seg<6>$ ).

Con esto podremos elegir los valores de entrada (a y b) a través de los conmutadores y ver el resultado en el visualizador 7 segmentos.

- 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 simplificado para la placa Basys2 que tiene todas las líneas comentadas. Nosotros únicamente quitaremos los comentarios y configuraremos las líneas correspondientes a las entradas y salidas escribiendo sus nombres cuando sea necesario.

**En este caso será necesario asociar  $a[3:0]$  y  $b[3:0]$  a los  $sw<i>$  correspondientes,  $op[1:0]$  a BTN1 y BTN0 y finamente descomentar las líneas correspondientes a  $seg<i>$ ,  $Led<i>$ ,  $an<i>$  y  $dp$  puesto que los nombres de esas señales coinciden con las definidas en el diseño.**

- Salve el fichero UCF.

## 4.5 Síntesis e Implementación

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

### 4.5.1 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 *calculadora.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 independiente *Adept* suministrada por el fabricante de la placa Basys2. Disponible en MS-Windows.
2. Emplear la herramienta de configuración del entorno ISE (iMPACT). Disponible en MS-Windows y GNU/Linux.
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.
- Pruebe el diseño

## 5 Cuarta parte: proyectos adicionales

---

En esta parte el alumno deberá realizar la simulación e implementación en la FPGA de los siguientes circuitos:

- Proponga el diseño de una unidad aritmético lógica con las operaciones que considere.