

Apellidos, Nombre: _____

Subsistemas Combinacionales

*Circuitos Electrónicos Digitales
Ingeniería Informática. Tecnologías Informáticas
Dpto. de Tecnología Electrónica
Septiembre, 2020*

1 Descripción y objetivos

En esta práctica tiene como principal objetivo el trabajo con subsistemas combinacionales de propósito general y específico incluyendo los circuitos aritméticos. Para ello se considerarán dos posibilidades:

- El uso de Circuitos integrados de escala de integración pequeña y mediana (SSI-Small Scale Integration, MSI-Medium Scale Integration).
- El uso de dispositivos programables.

Se trabajan los siguientes conceptos y competencias:

- Diseño de circuitos combinacionales haciendo uso de subsistemas.
- Diseño de circuitos aritméticos.
- Implementación mediante elementos discretos.
- Uso del instrumental habitual de laboratorio de electrónica.
- Uso de lenguajes de descripción de hardware (en concreto, Verilog) para simulación de circuitos combinacionales y subsistemas aritméticos.
- Uso de Verilog para implementar un circuito 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

- Instrumentación del laboratorio de Electrónica Digital.
- Circuitos integrados
- 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.
- Manual del instrumental básico de laboratorio de la asignatura (<https://www.dte.us.es/docencia/etsii/gii-ti/cedti/laboratorio>).
- Temas de subsistemas combinacionales y circuitos aritméticos.
- Diagrama de pines de los circuitos integrados. Documento patillaje en <https://www.dte.us.es/docencia/etsii/gii-ti/cedti/laboratorio>

3 Primera parte: Simulación

Actividades a realizar:

1. Estudiar el diseño de los circuitos descritos en el apartado 3.1.
2. Realizar la descripción Verilog del comparador diseñado mediante decodificador.
3. Realizar la descripción Verilog del comparador diseñado mediante multiplexor.
4. Realizar la descripción Verilog de la calculadora simple.
5. Utilizar el testbench para simular el comportamiento de los circuitos descritos.

3.1 Diseño en papel de los circuitos a simular

En este apartado se explicará como diseñar los circuitos que deben ser descritos y simulados en verilog

3.1.1 Diseño de comparador mediante subsistemas

Descripción verbal:

Diseñe un comparador de dos números de dos bits que nos indique en su única salida si los números son iguales. Realice el diseño de dos formas:

1. con un **decodificador** con salidas activas en bajo y una puerta **NAND**;
2. con un **multiplexor** de 8 canales e **inversores**.

Paso 1.

Se realiza el diagrama de bloques del circuito a diseñar y se especifican adecuadamente las entradas y salidas al mismo (figura 1).

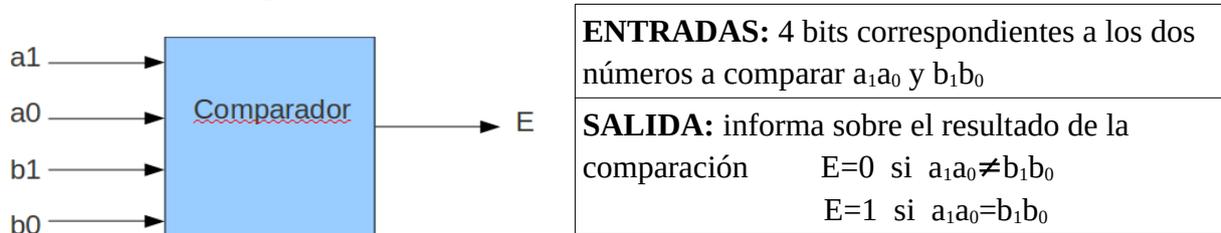


Figura 1: Diagrama de bloques

Paso 2.

Se representa la función de conmutación correspondiente a la descripción verbal. En nuestro caso utilizaremos un k-mapa (tabla 1).

Tabla 1. K-mapa

$a_1 a_0$	$b_1 b_0$	00	01	11	10
00	1	0	0	0	0
01	0	1	0	0	0
11	0	0	1	0	0
10	0	0	0	1	0
		E			

Paso 3.

Implementación del circuito

1. Mediante un **decodificador** con salidas activas en bajo y una puerta **NAND**.

Debemos tener en cuenta que este tipo de decodificador genera a sus salidas todas las posibles sumas de las variables que tengamos conectadas a sus entradas. El diseño consiste en conectar las salidas adecuadas del decodificador a las entradas de la puerta NAND.

La operación NAND de estas sumas nos permitirá obtener la función como suma de productos, esto debe ser tenido en cuenta a la hora de determinar las sumas que nos interesa conectar. El diseño se muestra en la Figura 2.

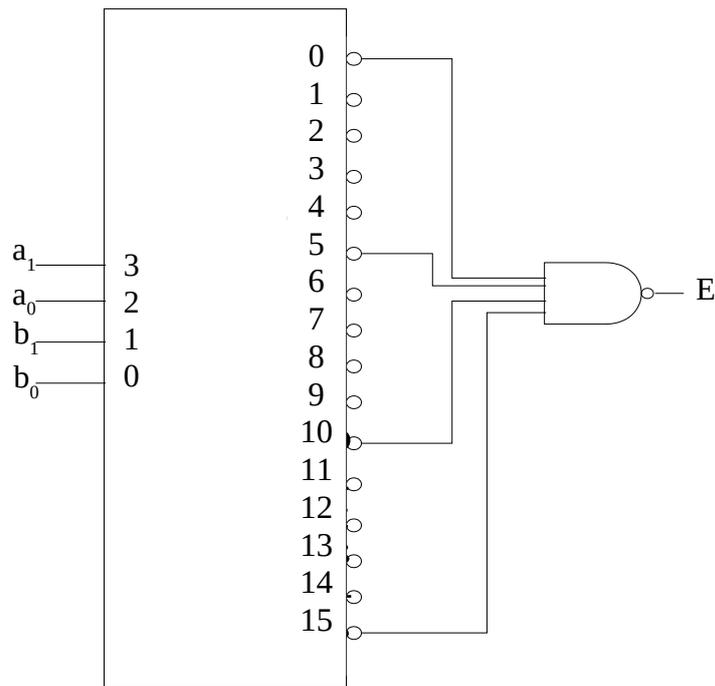


Figura 2. Diseño con subsistema de propósito específico

2. Con un **multiplexor** de 8 canales e **inversores**.

Elegimos a_1 , a_0 y b_1 como variables de selección. Señalemos para este caso los residuos sobre el mapa

$$\begin{array}{ll}
 E(0,0,0,b_0) = \overline{b_0} & E(1,0,0,b_0) = 0 \\
 E(0,0,1,b_0) = 0 & E(1,0,1,b_0) = \overline{b_0} \\
 E(0,1,0,b_0) = b_0 & E(1,1,0,b_0) = 0 \\
 E(0,1,1,b_0) = 0 & E(1,1,1,b_0) = b_0
 \end{array}$$

$a_1 a_0$	$b_1 b_0$	00	01	11	10
00		1			
01			1		
11				1	
10					1
		E			

De esta manera sabemos los valores que hay que conectar a las entradas de datos del MUX quedando el circuito que se muestra en la figura 3.

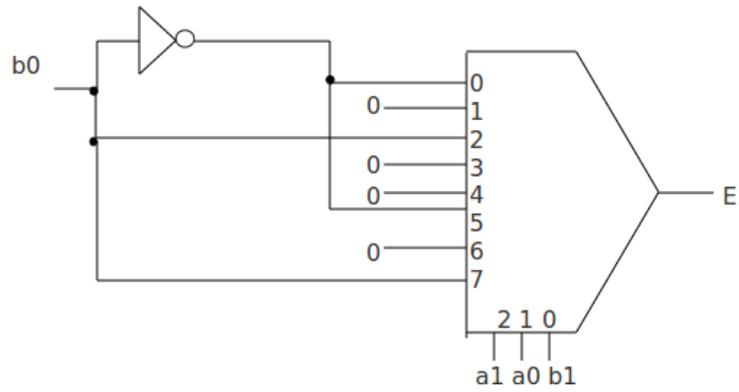


Figura 3. Diseño con subsistema de propósito general

3.1.2 Diseño de la unidad aritmética

Diseñe un circuito aritmético que realice las operaciones que se indican en la tabla 2.

Tabla 2. Operaciones de la UA

OP ₁ OP ₀	F
0 0	a + b
0 1	a – b
1 0	a
1 1	b

En la figura 4 se muestra el diseño de una etapa de este circuito.

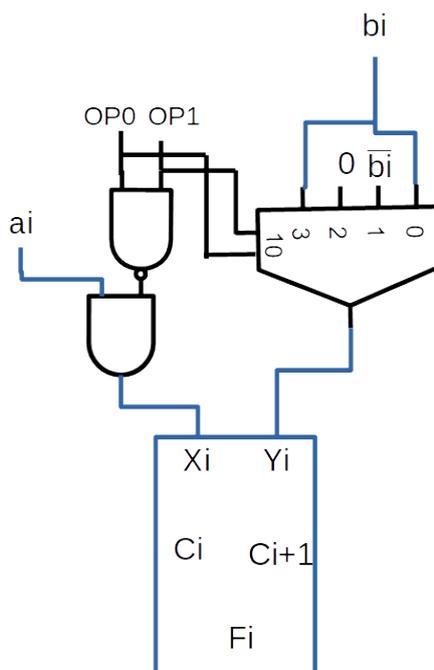


Figura 4. Etapa del circuito aritmético

Si se considera un diseño para números de 4 bits se replicaría 4 veces esta etapa.

A la hora de realizar la descripción Verilog de este circuito no se considerará sin embargo una descripción estructural, ya que es mucho más simple la procedimental. Por lo tanto nos basaremos en la tabla que nos muestra la descripción funcional.

3.2 Descripción Verilog del comparador

En este apartado se describen los pasos a seguir para realizar las simulaciones del comparador.

3.2.1 Descripción Verilog del comparador con decodificador

Pasos a seguir:

1. Estudie la descripción Verilog proporcionada del decodificador y pruebe su funcionamiento haciendo uso del testbench.
2. Complete la plantilla para el comparador, para ello deberá instanciar el decodificador y realizar la operación OR de sus salidas.
3. Realice el testbench del comparador.

3.2.2 Descripción Verilog del comparador con multiplexor

Pasos a seguir:

1. Estudie la descripción Verilog proporcionada del multiplexor y pruebe su funcionamiento haciendo uso del testbench.
2. Complete la plantilla para el comparador, para ello deberá instanciar el multiplexor proporcionándole las entradas adecuadas.
3. Realice el testbench del comparador.

3.3 Descripción Verilog de la Unidad Aritmética

1. Complete la descripción Verilog de la unidad aritmética.
2. Compruebe su funcionamiento mediante el testbench.
3. Revise la descripción Verilog del convertidor 7 segmentos.
3. Complete la descripción Verilog de la calculadora y revise la sintaxis.

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

Se deberá entregar un documento que contendrá:

- Descripciones Verilog de los circuitos propuestos.
- Resultados obtenidos en la simulación y explicación de los mismos

4 Segunda Parte: Montaje del comparador mediante CIs

Montaje del circuito de la figura 3 utilizando un CI 74151 y un 7404.

1. Indique las conexiones realizadas sobre la la plantilla de la figura 5. Tenga en cuenta que el mutiplexor tiene una señal de habilitación activa en bajo denominada STROBE.

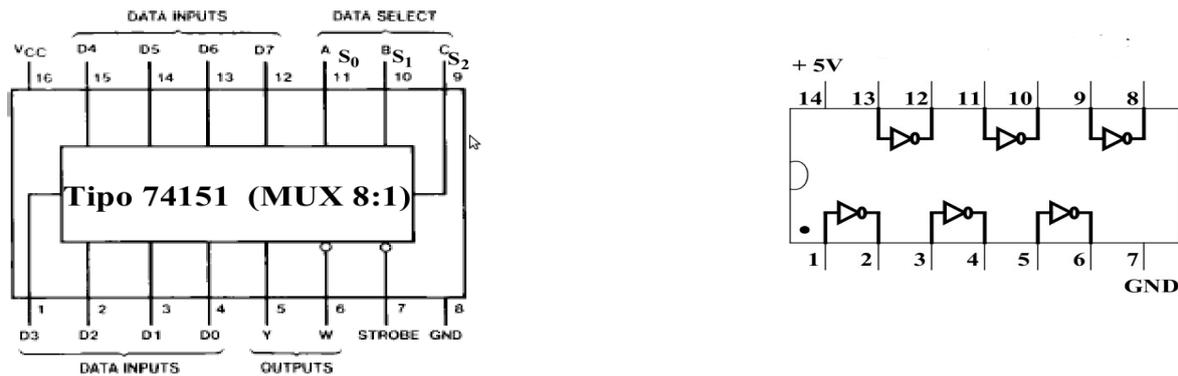


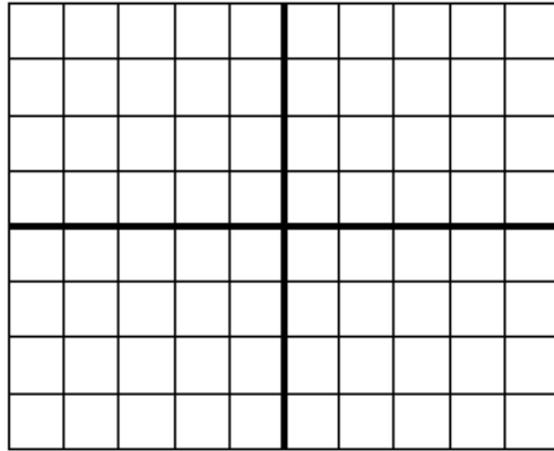
Figura 5 Plantilla para montaje 2

2. Compruebe el funcionamiento correcto de dicho circuito. Para ello siga el procedimiento habitual: cambie manualmente las entradas a_1 , a_0 y b_1 y conecte a b_0 una señal cuadrada que varíe entre 0V y 5V.
3. Rellene la tabla 3, indicando los posibles valores de E (0,1, b_0 , b_0').

Tabla 3

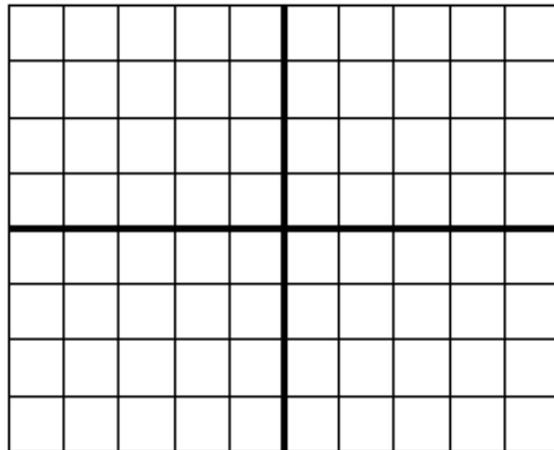
$a_1 a_0 b_1$	E
0 0 0	
0 0 1	
0 1 0	
0 1 1	
1 0 0	
1 0 1	
1 1 0	
1 1 1	

4. Dibuje en los gráficos siguientes los casos $a_1a_0b_1 = 001$ y $a_1a_0b_1 = 101$.



Escala de tiempo: _____

Escala de tensión: _____

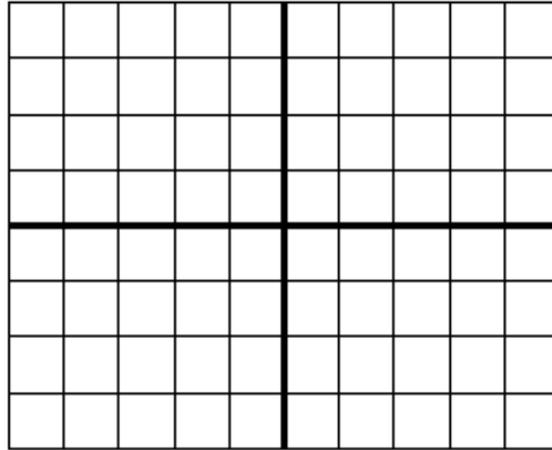


Escala de tiempo: _____

Escala de tensión: _____

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

5. Mida el tiempo de retardo del circuito para el caso $a_1a_0b_1 = 111$ y dibuje lo que observa en el osciloscopio cuando realiza dicha medida.



Escala de tiempo: _____

Escala de tensión: _____

$t_{pHL} =$ _____

$t_{pLH} =$ _____

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

5 Tercera parte: Implementación de la calculadora en FPGA

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

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 “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.

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. 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 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 la calculadora mapearemos las entrada en los switches de la placa y también en los LEDs para que nos permita la visualización de los valores que hemos puesto y la salidas z en uno de los visualizadores 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 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.
- Salve el fichero UCF.

5.4 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

5.4.1 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 `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 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 `dj tgcfg` 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

6 Cuarta parte: proyectos adicionales

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

- Circuito comparador. Proponga un circuito comparador mediante una descripción directa y mucho más sencilla que las desarrolladas en la práctica.
- Proponga el diseño de una Unidad aritmético lógica con las operaciones que considere.