

# Práctica 2 – Descripción comportamental y estructural de sistemas digitales

El objetivo de esta práctica es terminar de familiarizarse con el entorno Vivado de Xilinx y la placa de prototipado Nexys-4 DDR de Digilent, y profundizar en el modelado de sistemas digitales por medio de VHDL.

Se pretende desarrollar un sumador combinacional de N bits con propagación de acarreo serie (descripción estructural) y visualizar, simultáneamente, los datos de entrada, suma y acarreo resultantes a través de los displays de 7 segmentos (descripción FSM [*Finite State Machine*] comportamental).

## 1. Creación de un sumador completo

Al igual que en la Práctica 1, es posible obtener las ecuaciones lógicas de un sumador completo (Ecuación 1) a partir de su tabla de verdad (Tabla 1), y obtener así su estructura en base a puerta lógicas (Figura 1).

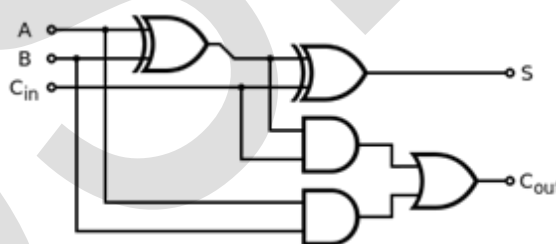
**Tabla 1. Tabla de verdad de un sumador completo.**

a	b	Cin	Cout	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$s = a \oplus b \oplus C_{in}$$

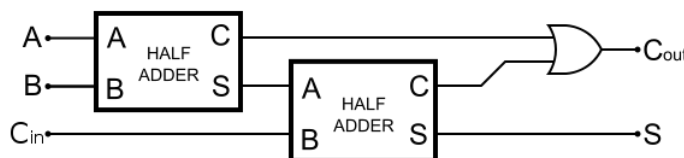
$$C_{out} = (a \cdot b) + C_{in} \cdot (a \oplus b)$$

**Ecuación 1. Funciones lógicas de un sumador completo.**



**Figura 1. Diseño estructural de un sumador completo.**

Sin embargo, tal y como muestra la Figura 2, es posible construir un sumador completo a partir del semisumador diseñado en la Práctica 1. Así pues, se realizará el diseño estructural de este componente en base al semisumador diseñado anteriormente.



**Figura 2. Diseño estructural de un sumador completo a partir de semisumadores.**

Este diseño deberá simularse para verificar su correcta descripción.

## 2. Creación de un sumador completo de N bits con propagación de acarreo serie

El sumador completo de 1 bit recién diseñado es la base para la construcción de sumadores completos con propagación de acarreo serie. Tal y como muestra la Figura 3, cada sumador completo realiza la suma del bit  $i$ -ésimo de las entradas junto con el acarreo del sumador completo  $i-1$ . El acarreo resultante se pasa como acarreo de entrada al sumador completo  $i+1$ .

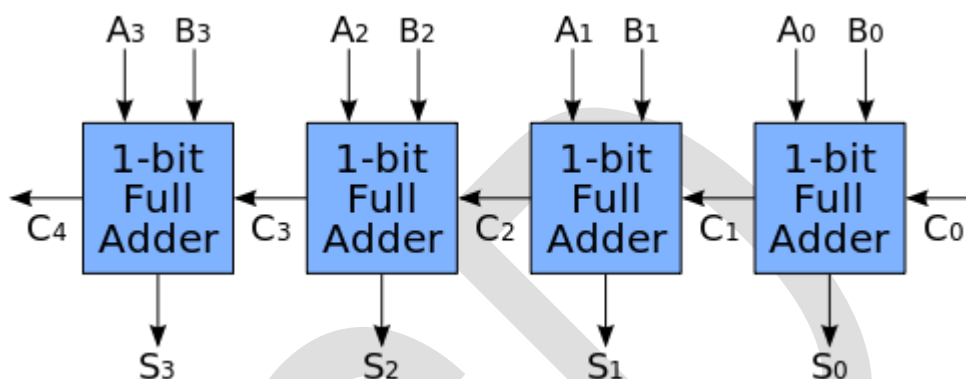


Figura 3. Diseño estructural de un sumador completo con propagación de acarreo serie.

Esta estructura es fácilmente traducible a un modelo VHDL en el que se utilicen elementos genéricos para parametrizar la generación del componente deseado.

Este diseño deberá simularse para verificar su correcta descripción.

## 3. Creación de un controlador de visualización en displays de 7 segmentos

Tal y como se indica en la página 19 del manual de referencia de la placa de prototipado, los 8 displays de 7 segmentos disponibles en la placa comparten la misma entrada de datos. Esto significa que el mismo número se mostrará en todos los displays que tengan su entrada de habilitación activa.

Para poder visualizar un número diferente en cada uno de estos displays, de manera *aparentemente simultánea*, es necesario multiplexar su utilización en el tiempo. Un ejemplo de cómo realizar esta multiplexación a través de una pequeña máquina de estados finitos (*FSM – Finite State Machine*) se muestra en la figura disponible en la página 20 del manual y en la Figura 4. Si el barrido de los displays se realiza a una velocidad lo suficientemente rápida, el ojo humano tendrá la sensación de que todos los displays están activos simultáneamente mostrando cada uno de ellos un número diferente. Sin embargo, debe realizarse a una velocidad que permita a los displays apagarse completamente, o se percibirá como si todos los displays estuvieran semi-encendidos. De acuerdo al manual de referencia, **el refresco de los 4 displays debe realizarse entre una vez cada 1 ms y una vez cada 16 ms.**

*La máquina de estados deberá utilizar la misma señal de reloj que el resto del sistema, y dispondrá de una señal de habilitación que le permita permanecer en el mismo estado, por tiempo indefinido, hasta que dicha señal se active.*

Este diseño deberá simularse para verificar su correcta descripción.

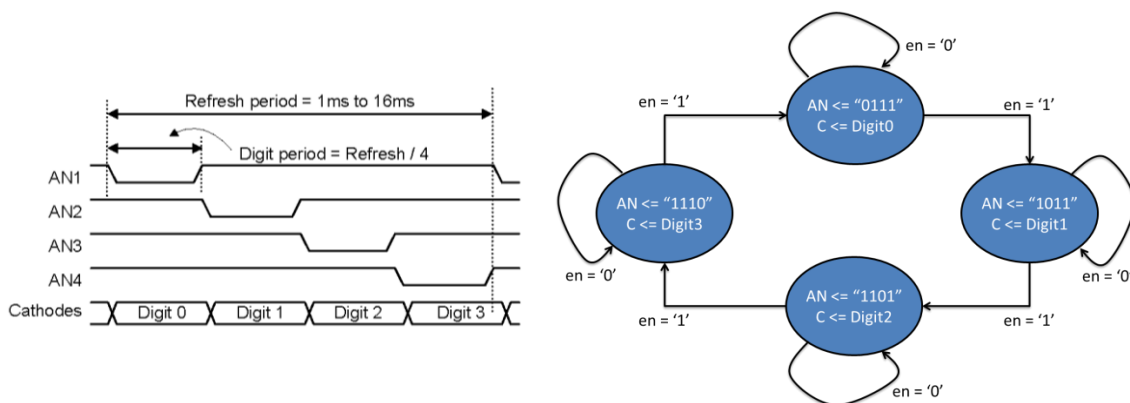


Figura 4. Máquina de estados para la multiplexación en el tiempo de los 4 displays de 7 segmentos.

## 4. Creación de un divisor de frecuencia

Según la información disponible en la página 9 del manual de referencia, la placa de prototipado dispone de un oscilador CMOS (reloj) de 100 MHz conectado al pin E3 de la FPGA. Evidentemente, la frecuencia de esta señal de reloj es demasiado elevada para el correcto funcionamiento del controlador de los displays de 7 segmentos, ya que los displays se encenderían/apagarían alrededor de 25M de veces por segundo.

Para limitar la frecuencia de activación de los leds (controlador del display de 7 segmentos), es necesario desarrollar un circuito que reduzca la frecuencia de funcionamiento del sistema, conocido como *divisor de frecuencia*. Básicamente, este componente generará un pulso en su salida cada vez que detecte un número determinado de pulsos de reloj. Este pulso permitirá activar los componentes deseados una vez cada X ciclos de reloj. Así, por ejemplo, un divisor de reloj por 4, generará un pulso en su salida cada 4 pulsos de reloj recibidos, por lo que estará generando una señal cuya frecuencia será la cuarta parte de la frecuencia original. Un cronograma de ejemplo se muestra en la Figura 5.

**IMPORTANTE:** Esta señal se utilizará para controlar la *señal de habilitación* de los componentes que tengan que funcionar a una frecuencia reducida, y todos ellos utilizarán *la señal de reloj original*. Con ello se evita el problema de generar diversos dominios de reloj.

El diseño deberá ser genérico, para poder utilizar el divisor de frecuencia en otros proyectos, y deberá simularse para verificar su correcta descripción.

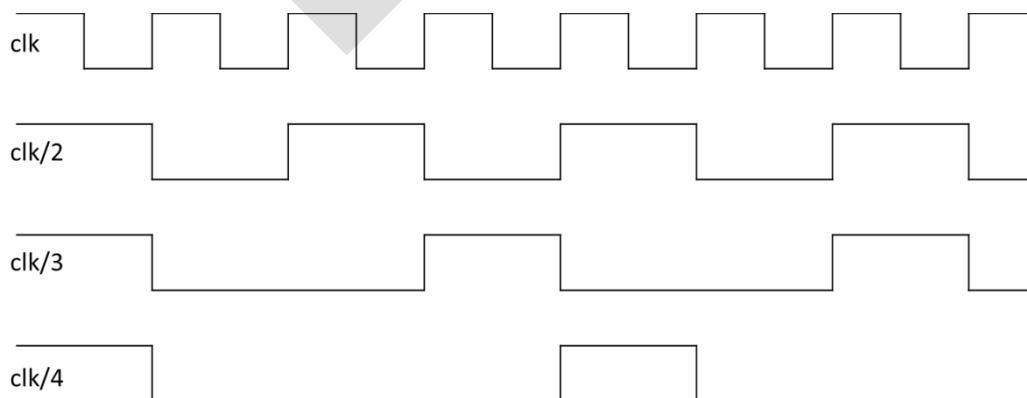


Figura 5. Salida de divisores de frecuencia por 2, 3 y 4 en función de la entrada de reloj (clk).

## 5. Integración de todos los componentes

El sistema final deberá tomar los dos números de 4 bits a sumar a través de los interruptores disponibles, y mostrará dichos números, junto con la suma y el acarreo, utilizando los displays de 7 segmentos. Un ejemplo de integración de todos los componentes para la implementación del sistema se muestra en la Figura 6.



Figura 6. Ejemplo de utilización del sistema a desarrollar.

Se pide verificar el correcto funcionamiento del sistema diseñado en la placa de prototipado.

## 6. **Bonus:** Creación de un sumador de N bits por medio de una descripción comportamental

Realizar la descripción comportamental de un sumador con la misma interfaz que el sumador con propagación de acarreo serie de N bits.

Se pide sustituir el sumador estructural por la nueva versión comportamental y verificar su correcto funcionamiento en la placa de prototipado.