

Escuela Superior de Ingenieros

Universidad de Sevilla
Departamento de Ingeniería Electrónica

Primera práctica de VHDL

Microelectrónica

Jonathan N. Tombs

Fernando Muñoz Chavero

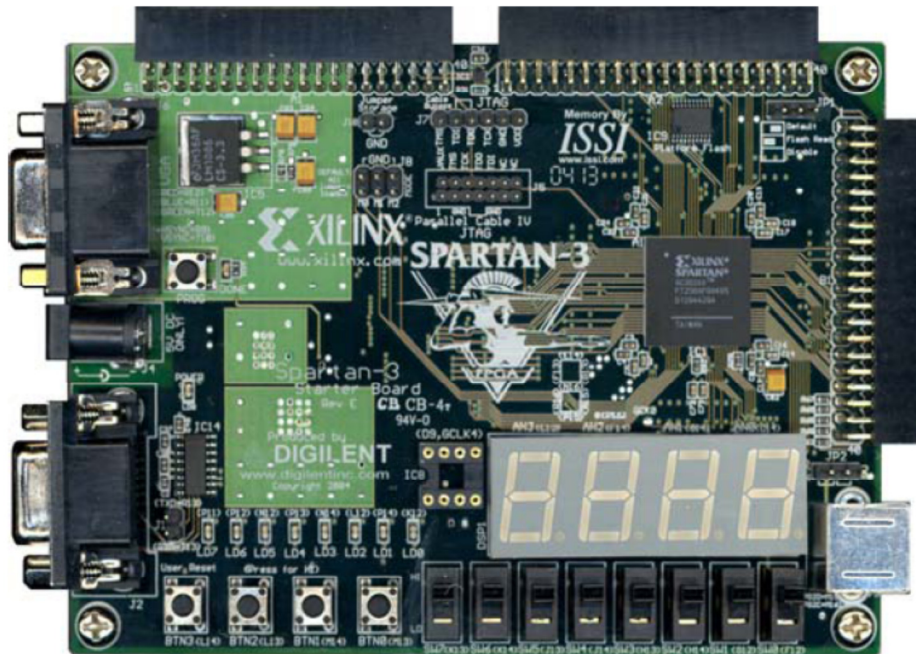
Miguel A. Aguirre

Hipólito Guzmán

Javier Nápoles

(Versión: febrero de 2013)

Práctica 1. Control del Display de 7 segmentos



Práctica 1. Control del Display de 7 segmentos

Esta práctica es la primera de un grupo de tres dedicada a entrenar al alumno con el entorno de trabajo Xilinx ISE y la tarjeta Digilent S3.

Se diseñarán una serie de contadores síncronos que permitirán controlar los cuatro displays de 7 segmentos disponibles en la tarjeta. Los principales objetivos didácticos son los siguientes:

- 1. Aprender las opciones básicas de simulación e implementación del entorno Xilinx ISE.*
- 2. Aprender a utilizar la hoja de características de la tarjeta S3 para asignar correctamente los puertos de la FPGA en función de la aplicación.*

1. Objetivo propuesto

La tarjeta Digilent S3 de Xilinx es una placa de desarrollo para la FPGA de Xilinx Spartan-3. Dicha tarjeta dispone de múltiples dispositivos de entrada/salida conectados directamente a la FPGA. En la presente práctica nos centraremos en la utilización del display LED de 7 segmentos y cuatro dígitos.

El objetivo es convertir el display en un contador que se incremente a una velocidad “visible”. Para ello necesitaremos diseñar:

- a. Primera parte de la práctica:
 - Un contador de 26 bits que para dividir la frecuencia de reloj.
 - Un contador entre 0 y 9 con reset síncrono y señal de saturación.
 - Un decodificador de binario a código de 7 segmentos.
- b. Segunda parte de la práctica:
 - Pequeño registro de desplazamiento que controlará la señal de habilitación de cada dígito.

2. Tarjeta S3 de Digilent

Para la realización de esta práctica utilizaremos la tarjeta S3 de Digilent. En este apartado se explicará cómo funcionan los recursos de la tarjeta que utilizaremos en la práctica¹.

Display de 7 segmentos y cuatro dígitos

Básicamente, la tarjeta Digilent S3 consiste en una FPGA (Xilinx Spartan-3 con 200K puertas equivalentes) cuyos puertos están unidos directamente a una serie de periféricos. En la Figura 1 se muestra el diagrama de bloques básicos de la tarjeta.

Los cuatro displays de 7 segmentos están situados en la parte inferior derecha de la tarjeta. En el control de los displays, explicado en la Figura 2, se utilizarán las siguientes señales (conectadas a puertos de la FPGA):

- *A, B, C, D, E, F, G, y DP*: 8 señales comunes para los cuatro dígitos activas a nivel bajo. Cuando una de estas señales se activa (fijando un ‘0’ desde la FPGA) se encenderá el segmento correspondiente.

¹ Toda la información de este apartado se obtenido del documento [Spartan-3 Starter Kit Borrador User Guide](#) disponible en la página de Digilent. Se anima al alumno a utilizar el documento original.

- AN0, AN1, AN2 y AN3: Señal de habilitación (activa a nivel bajo). Tendremos una habilitación independiente para cada dígito, de forma que sólo se encenderá el que esté habilitado.

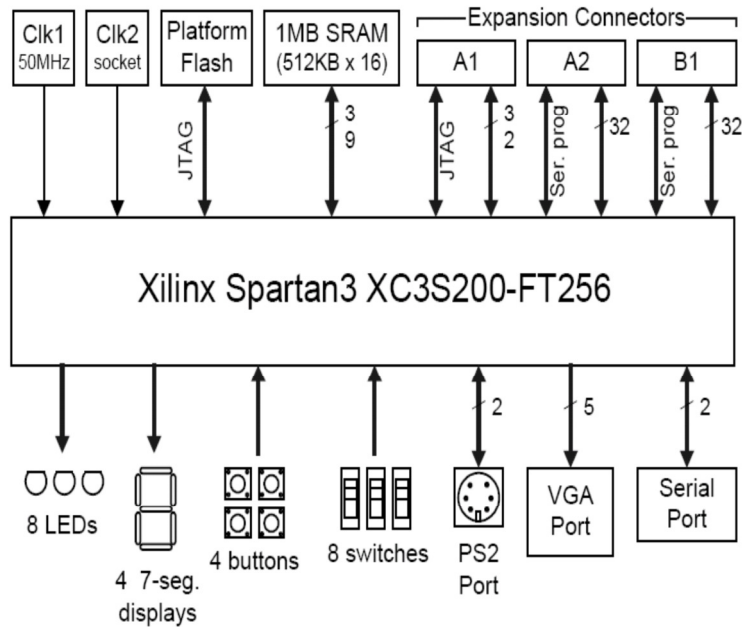


Figura 1. Diagrama de bloques de la tarjeta S3

A modo de ejemplo, en la Figura 2, se ilustra qué estímulos (en azul) habría que proporcionar a los displays para conseguir que aparezca un 2 en el primer dígito:

- Se activan (valor lógico '0') las entradas A, B, E, D y G. Se dejan desactivadas todas las demás.
- Se selecciona sólo el primer dígito (AN3='0'), manteniendo desactivados los demás (AN0=AN1=AN2='1').

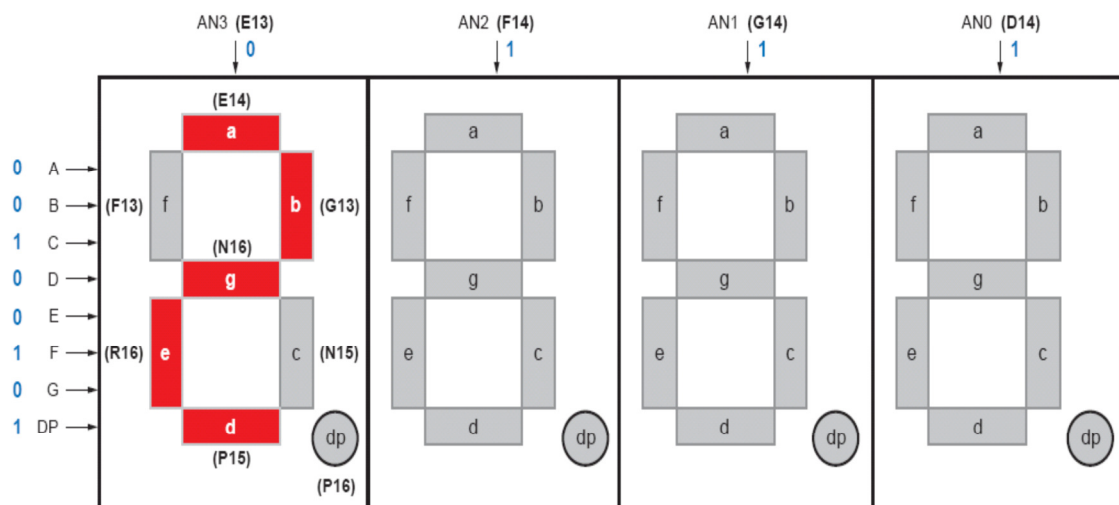


Figura 2. Control de los displays de segmentos

Es muy sencillo deducir (Tabla 1) el código de entrada necesario para representar un carácter en el display. Esta tabla se utilizará para diseñar un decodificador de binario a 7-segmentos.

Character	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
b	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
d	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

Tabla 1. Códigos necesarios para la representación de caracteres

Un aspecto muy importante a la hora de implementar un diseño en la tarjeta es selección de puertos en la FPGA. En función de los periféricos o conectores que vayamos a utilizar en nuestro proyecto utilizaremos unos puertos concretos de la FPGA, que deberán ser proporcionados al programa ISE al realizar la implementación. Para determinar los puertos de nuestro diseño necesitamos conocer cómo está conectada la FPGA con los diferentes periféricos en la tarjeta S3.

En Tabla 2 se especifica cómo está conectada la FPGA y los displays de 7 segmentos. El diseño final de esta práctica, una vez implementado en la FPGA, debe controlar los pines del display a través de estos puertos.

Segment	FPGA Pin			
A	E14			
B	G13			
C	N15			
D	P15			
E	R16			
F	F13			
G	N16			
DP	P16			

Anode Control	AN3	AN2	AN1	AN0
FPGA Pin	E13	F14	G14	D14

Tabla 2. Conexión entre la FPGA y los displays de 7 segmentos (activos a nivel bajo)

En este punto podríamos hacernos una pregunta. Si los cuatro displays comparten los 8 bits que encienden cada uno de los segmentos, ¿cómo podemos utilizar los cuatro displays independientemente? La respuesta a esta pregunta está ilustrada en la Figura 3.

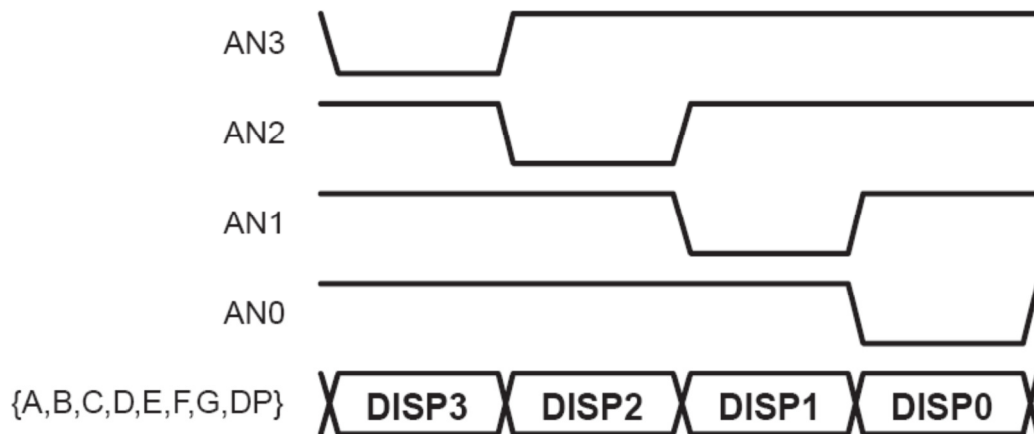


Figura 3. Utilización de los 4 dígitos simultáneamente

Las señales de habilitación de los displays (AN0-3) servirán para multiplexar en el tiempo las señales de activación de los segmentos (A,B, ..., DP). De esta forma estamos encendiendo sucesivamente un display y apagando todos los demás. Debido a la persistencia de la visión, el cerebro humano percibe que todos los caracteres se encienden simultáneamente.

Señal de reloj

La tarjeta S3 dispone de un reloj de 50MHz (Epson SG-8002JF) y de un zócalo para un generador de reloj opcional. En el laboratorio utilizaremos siempre la entrada de reloj de 50 MHz disponible en el **puerto T9** de la FPGA. Si fuera necesario utilizar otras frecuencias de reloj se podrían obtener a partir de los 50MHz utilizando el [Digital Clock Manager](#) de la Spartan 3.

Pulsadores y reset asíncrono

Para introducir el reset asíncrono a nuestro diseño utilizaremos uno de los cuatro pulsadores de la parte inferior izquierda de la tarjeta. Debajo de los pulsadores podemos encontrar una etiqueta con un nombre asignado al mismo (BTN3-0), y en la Tabla 3 se especifican los puertos de la FPGA conectados a los mismos. Al pulsar un botón producimos un nivel lógico '1' en el puerto correspondiente de la FPGA².

Push Button	BTN3 (User Reset)	BTN2	BTN1	BTN0
FPGA Pin	L14	L13	M14	M13

Tabla 3. Puertos asociados a los pulsadores

Utilizaremos como reset asíncrono activo a nivel alto el pulsador BTN3 conectado al puerto L14 de la FPGA.

3. Realización de la práctica

3.1. Primera parte

El objetivo de la primera parte de la práctica será diseñar un circuito que realice una cuenta del 0 al 9 en el primero de los cuatro displays de 7 segmentos disponible en la placa. Podemos distinguir los siguientes bloques a diseñar en VDDL:

- 1) DIV_FREQ: Un contador de 26 bits para dividir la frecuencia de reloj a una frecuencia *más visible por el ojo humano*.
- 2) CONT_DIGITO: Un contador de 0 a 9.
- 3) DECODIFICADOR: Un decodificador de binario a siete segmentos.

A continuación se explicará en detalle la funcionalidad de cada uno de estos bloques. Se recomienda al alumno realizar la simulación de la funcionalidad de cada uno de estos bloques antes de pasar al diseño del siguiente.

² Normalmente, para utilizar los pulsadores en un proyecto será necesario implementar un filtro de *glitches* en la FPGA. En la presente práctica, que utilizaremos el pulsador como reset asíncrono, no es necesario.

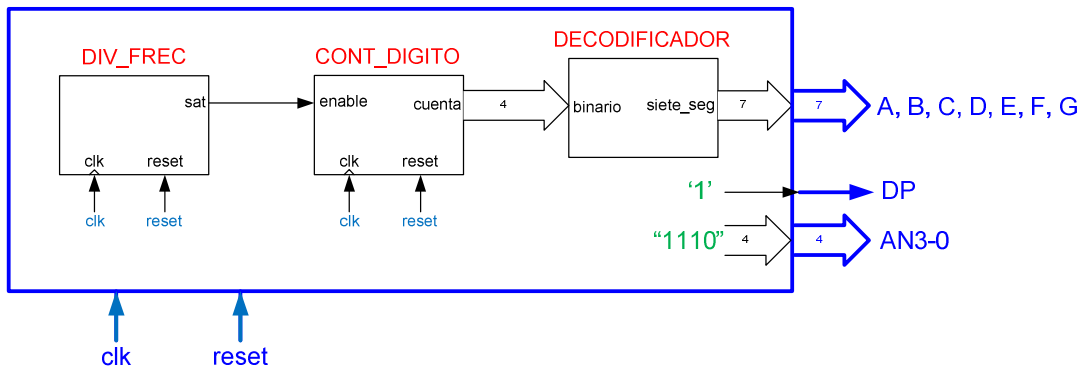


Figura 4. Diagrama de bloques para la primera parte de la práctica

Divisor de frecuencia

DIV_FREQ	
Descripción	<p>En la tarjeta S3 tenemos un reloj de 50MHz. Este bloque dividirá la esta frecuencia por 2^{26} para obtener una señal a una frecuencia de 0.74Hz.</p> <ul style="list-style-type: none"> El bloque consiste en un contador de 26bits cuyo valor de cuenta se almacenará internamente en un registro de 26 bits denominado <i>cuenta</i>. En cada flanco de subida del reloj el contador incrementará su valor en uno. En la salida <i>sat</i> se proporcionará un pulso de reloj cada vez que <i>cuenta</i>=67108863. (es equivalente a <i>cuenta</i>="111111111111111111111111").³
Entidad ⁴	<pre>entity div_freq is Port (clk : in STD_LOGIC; reset : in STD_LOGIC; sat : out STD_LOGIC); end div_freq;</pre>
Descripción de los puertos	
clk	Reloj
reset	Reset asíncrono activo a nivel alto
sat	Se produce un pulso, <u>de duración un solo ciclo de reloj</u> , cada vez que el contador llega la final de la cuenta: <i>cuenta</i> =67108863 (o análogamente <i>cuenta</i> ="111111111111111111111111")

³ Para una simulación más ágil se recomienda diseñar en primer lugar una versión del contador con cuatro bits para almacenar *cuenta*, de forma que produzca un pulso en *sat* cada vez que *cuenta*=15 (ó "1111").

⁴ Con objetivo de definir correctamente la interfaz del bloque se proporciona la entidad. El trabajo del alumno consiste en declarar las librerías correctas y escribir la arquitectura.

Contador de 0 a 9

CONT_DIGITO	
Descripción	Contador síncrono con habilitación. <ul style="list-style-type: none"> • Cuando está activa la habilitación se contará de 0 a 9, volviendo a comenzar de nuevo por 0.
Entidad	<pre>entity cont_digito is Port (clk : in STD_LOGIC; reset : in STD_LOGIC; enable : in STD_LOGIC; cuenta : out STD_LOGIC_VECTOR (3 downto 0)); end cont_digito;</pre>
Descripción de los puertos	
clk	Reloj
reset	Reset asíncrono activo a nivel alto
enable	Señal de habilitación activa a nivel alto. Si enable='1' el contador avanzará uno en la cuenta en el flanco positivo de reloj.
cuenta	Salida de cuatro bits. Comenzará en "0000" (0) y terminará en "1001" (9), volviendo a empezar de nuevo.

Decodificador de binario a 7 segmentos

DECODIFICADOR	
Descripción	Bloque combinacional que produce el código correcto para la activación del display de 7 segmentos a partir del número en binario de 4 bits ⁵ . El diseño de este bloque es trivial utilizando la Tabla 1 y una sentencia <i>case</i>
Entidad	<pre>entity decoficador is Port (binario : in STD_LOGIC_VECTOR (3 downto 0); siete_seg : out STD_LOGIC_VECTOR (6 downto 0)); end decoficador;</pre>
Descripción de los puertos	
binario	Código binario sin signo
siete_seg	Código de 7 segmentos según la Tabla 1. Se corresponde, de más a menos significativo, con las entradas A,B,C,D,E,F y G del display.

Bloque de jerarquía superior

Una vez diseñados y simulados los bloques anteriores pasaremos a conectarlos tal como se muestra en la Figura 4.

⁵ En esta práctica utilizaremos un bloque combinacional para atacar directamente los puertos de salida de la FPGA, pudiéndose producir *glitches* en esas salidas. Normalmente estos *glitches* son indeseables y para evitarlos (a partir de la siguiente práctica) produciremos todas las salidas de forma síncrona.

CONTROL_DISPLAY_P1	
Descripción	Realizará un contador de 4 dígitos en el display de 7 segmentos. Bloque de mayor jerarquía que llama a todos los anteriores como componentes.
Entidad	<pre>entity control_display_p1 is Port (clk : in STD_LOGIC; reset : in STD_LOGIC; A : out STD_LOGIC; B : out STD_LOGIC; C : out STD_LOGIC; D : out STD_LOGIC; E : out STD_LOGIC; F : out STD_LOGIC; G : out STD_LOGIC; DP : out STD_LOGIC; AN : out STD_LOGIC_VECTOR (3 downto 0)); end control_display_p1;</pre>
Descripción de los puertos	
clk	Reloj
reset	Reset asíncrono activo a nivel alto
A,B,C,D,E,F,G	Salidas activas a nivel bajo que encienden los leds correspondientes del display (ver Figura 2)
DP	Salida activa a nivel bajo que enciende el punto del display. Siempre estará desactivada.
AN	Habilitación de los diferentes dígitos del display. Se le asignará el valor "1110" para que se encienda sólo uno de los displays. El alumno puede probar con otros valores ("1011", "1100", ...).

3.2. Segunda parte

En la segunda parte de la práctica se utilizarán todos los displays simultáneamente con objeto de realizar una cuenta de 0000 a 9999. En la Figura 5 se muestra el esquema general de la segunda parte de la práctica. Podemos distinguir los siguientes bloques a diseñar en VHDL:

- 1) DIV_FREQ: Un contador de 26 bits para dividir la frecuencia de reloj a frecuencia *más visible por el ojo humano*. Se realizará una pequeña modificación al divisor diseñado en la primera parte para que proporcione también una señal de 195KHz necesaria para la conmutación entre los displays.
- 2) CONT_DIGITO: Un contador de 0 a 9. Asignaremos un contador de este tipo a cada dígito del display. Añadiremos una señal de saturación al contador diseñado en la primera parte.
- 3) DECODIFICADOR: Un decodificador de binario a siete segmentos. Será el mismo diseñado en la primera parte.
- 4) REG_DESP: Un pequeño registro de desplazamiento que controlará la señal de habilitación de cada dígito. La señal de habilitación estará controlada por la salida del bloque DIV_FREQ a 195KHz.

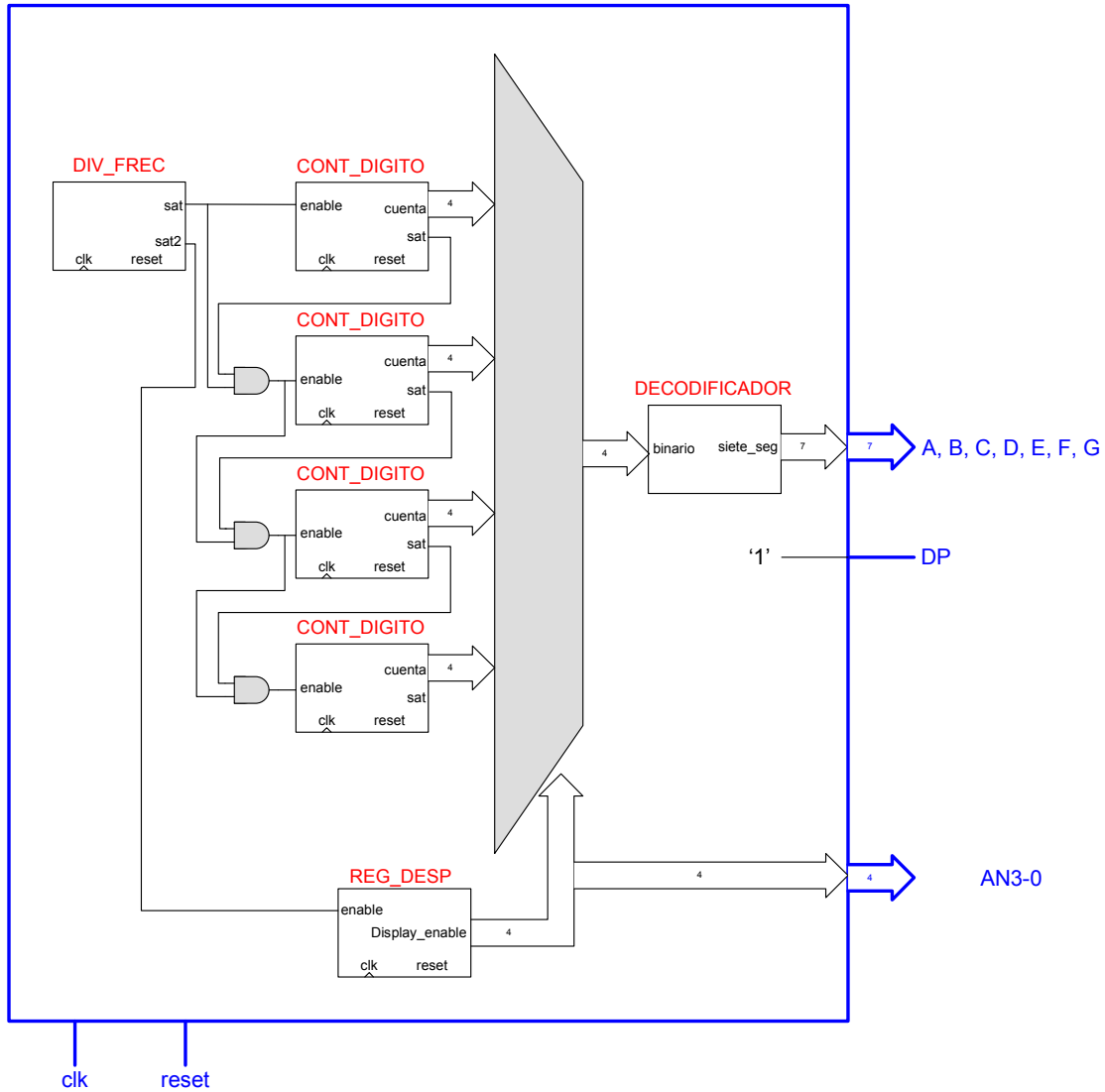


Figura 5. Diagrama de bloques para la segunda parte de la práctica

Los bloques sombreados de la Figura 5 (multiplexor y puertas AND) se implementarán directamente, mediante órdenes concurrentes, en el nivel de jerarquía superior.

Divisor de frecuencia

DIV_FREQ	
Descripción	<p>En la tarjeta S3 tenemos un reloj de 50MHz. Como en la primera parte de la práctica, este bloque dividirá esta frecuencia por 2^{26} para obtener una señal a una frecuencia de 0.74Hz (contando continuamente de 0 a 67108863).</p> <p>La única diferencia con el divisor diseñado en la primera parte es que se ha añadido una nueva salida de saturación, de forma que se producirán un pulso en:</p> <ul style="list-style-type: none"> • sat cada vez que <i>cuenta</i>=67108863 (o <i>cuenta</i>="11111111111111111111111111111111"). • sat2 cada vez que <i>cuenta</i>(7 downto 0) =255 (o <i>cuenta</i>(7 downto 0) ="11111111").
Entidad	<pre>entity div_freq is Port (clk : in STD_LOGIC; reset : in STD_LOGIC; sat : out STD_LOGIC; sat2 : out STD_LOGIC); end div_freq;</pre>
Descripción de los puertos	
Clk	Reloj
reset	Reset asíncrono activo a nivel alto
sat	Se produce un pulso, de duración un solo ciclo de reloj, cada vez que el contador llega la final de la cuenta: <i>cuenta</i> =67108863 (<i>cuenta</i> ="11111111111111111111111111111111")
Sat2	Se produce un pulso, de duración un solo ciclo de reloj, cada vez que los ocho bits menos significativos de <i>cuenta</i> llegan a: <i>cuenta</i> (7 downto 0) =255 (<i>cuenta</i> (7 downto 0) ="11111111")

Contador de 0 a 9

CONT_DIGITO	
Descripción	Contador síncrono con habilitación. Cuando está activa la habilitación se contará de 0 a 9, volviendo a comenzar de nuevo por 0. Añadiremos al contador diseñado en la primera parte una señal de saturación
Entidad	<pre>entity cont_digito is Port (clk : in STD_LOGIC; reset : in STD_LOGIC; enable : in STD_LOGIC; cuenta : out STD_LOGIC_VECTOR (3 downto 0); sat : out STD_LOGIC); end cont_digito;</pre>
Descripción de los puertos	
clk	Reloj
reset	Reset asíncrono activo a nivel alto
enable	Señal de habilitación activa a nivel alto. Si enable='1' el contador avanzará uno en la cuenta en el flanco positivo de reloj.
cuenta	Salida de cuatro bits. Comenzará en "0000" y terminará en "1001", volviendo a empezar de nuevo.
sat	Si cuenta es igual a "1001", sat estará activada (activa a nivel alto).

Registro de desplazamiento

REG_DESP	
Descripción	Registro de desplazamiento circular. El valor del registro al producirse un reset asíncrono será "1110", por tanto, la secuencia que se producirá será: "1110" → "1101" → "1011" → "0111" → "1110". (Ver Figura 3). Sólo se producirá un desplazamiento si módulo está habilitado. El valor de registro se utilizará para controlar las señales de habilitación de los displays: AN3:0.
Entidad	<pre>entity reg_desp is Port (clk : in STD_LOGIC; reset : in STD_LOGIC; enable : in STD_LOGIC; display_enable : out STD_LOGIC_VECTOR (3 downto 0)); end reg_desp;</pre>
Descripción de los puertos	
clk	Reloj
reset	Reset asíncrono activo a nivel alto
enable	Habilitación activa a nivel alto
display_enable	Valor del registro de desplazamiento. Con enable='1', se producirá un desplazamiento cada flanco positivo de reloj: <ul style="list-style-type: none"> Todos los bits de desplazarán una posición a la izquierda (hacia el BIT más significativo). El BIT menos significativo pasa a tener el valor del BIT más significativo.

Bloque de jerarquía superior

Una vez diseñados y simulados los bloques anteriores pasaremos a conectarlos tal como se muestra en la Figura 5.

CONTROL_DISPLAY_P2	
Descripción	Realizará un contador de 4 dígitos en el display de 7 segmentos. Bloque de mayor jerarquía que llama a todos los anteriores como componentes.
Entidad	<pre>entity control_display_p2 is Port (clk : in STD_LOGIC; reset : in STD_LOGIC; A : out STD_LOGIC; B : out STD_LOGIC; C : out STD_LOGIC; D : out STD_LOGIC; E : out STD_LOGIC; F : out STD_LOGIC; G : out STD_LOGIC; DP : out STD_LOGIC; AN : out STD_LOGIC_VECTOR (3 downto 0)); end control_display_p2;</pre>
Descripción de los puertos	
clk	Reloj
reset	Reset asíncrono activo a nivel alto
A,B,C,D,E,F,G	Salidas activas a nivel bajo que encienden los leds correspondientes del display.
DP	Salida activa a nivel bajo que enciende el punto del display. Siempre estará desactivada.
AN	Habilitación de los diferentes dígitos del display. Se formará una secuencia como se indica en la figura Figura 3.