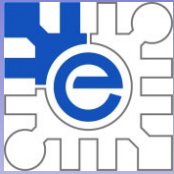




Universidad  
de Alcalá



Departamento  
de  
Electrónica

# Modelado de Sistemas Computacionales

Grado en Ingeniería de Computadores

## Práctica libre: Generador de señal controlado desde un puerto serie RS232

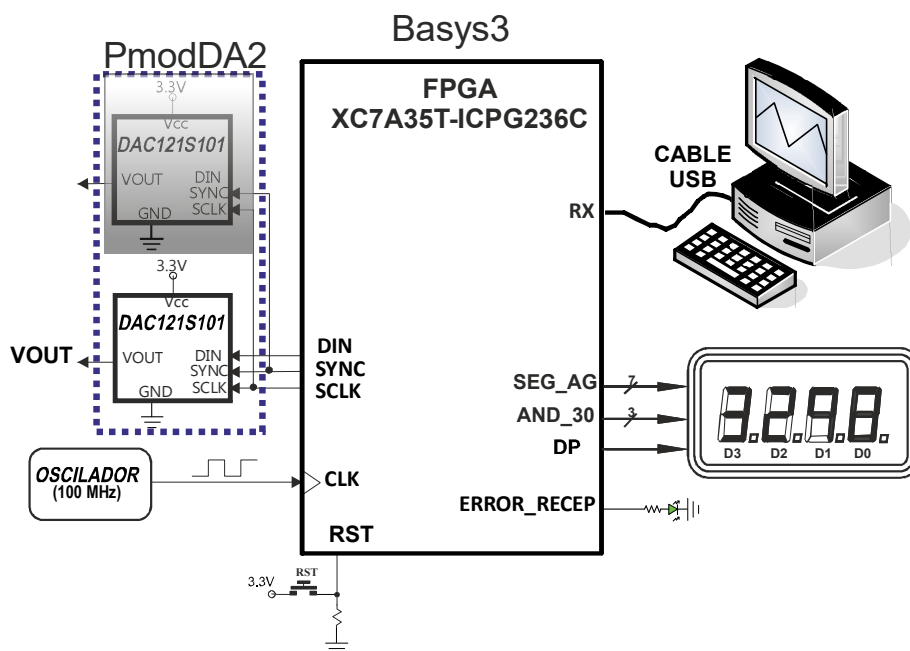
### Apartado 1.

#### Práctica Libre

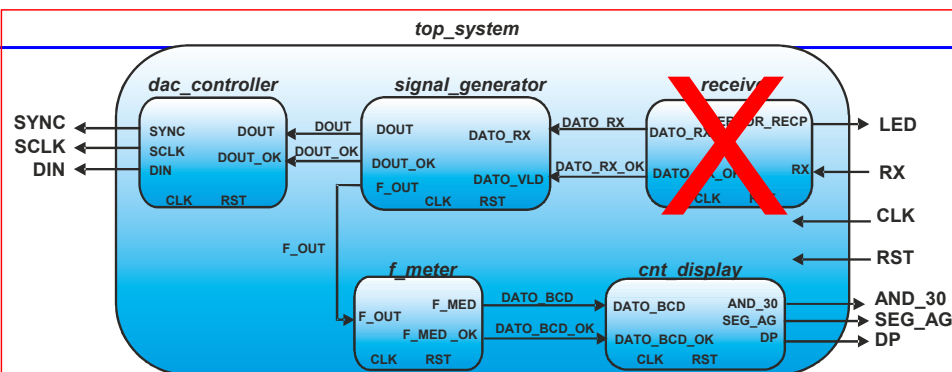
La nota final estará compuesta por las siguientes partes según la ponderación mostrada:

1. Examen teórico de aspectos prácticos de los diferentes apartados de la práctica. Esta prueba tiene un peso del **35%** de la nota final y se realizará el día **8 de mayo de 2023**.
2. Defensa individual de la práctica en la que el alumno deberá demostrar el conocimiento de las herramientas de diseño y de la práctica. Para ello, el alumno tendrá que realizar simulaciones funcionales y/o temporales de alguno de los apartados en un tiempo limitado y responder a las preguntas que se planteen en relación con el código VHDL utilizado para modelar los diferentes módulos. Esta prueba tiene un peso del **40%** de la nota final y se realizará el día **8 de mayo de 2023**.
3. Evaluación de los fuentes del diseño entregados (25% de la nota final).

**¡La nota del laboratorio se mantiene hasta la convocatoria extraordinaria!**



Modelar en VHDL de un sistema digital que permite, una vez programado en una FPGA, generar una señal analógica *VOUT* cuya forma de onda, amplitud y frecuencia se van a poder seleccionar desde un archivo creado y ejecutado en *Matlab*, estos valores serán enviados a través de una conexión RS232.



**NO SE PODRÁN MODIFICAR LOS PUERTOS DE CADA UNA DE LAS ENTIDADES PROPORCIONADAS**

**CLK:** entrada de sincronismo para los elementos secuenciales. Tiene un periodo  $T=10$  ns, frecuencia de 100 MHz.

**RST:** entrada activa a nivel alto, procedente de un pulsador de la placa de pruebas que se utiliza para inicializar, de forma asíncrona, los elementos secuenciales del diseño.

**RX:** entrada de datos proveniente de un ordenador y que utiliza el protocolo RS232.

**LED:** salida activa a nivel alto que indica que se ha producido un error en la recepción del dato serie.

**SYNC :** salida que indica que se está enviando un nuevo dato al DAC.

**SCLK:** salida de sincronismo utilizada por el DAC para almacenar los bits del dato a convertir.

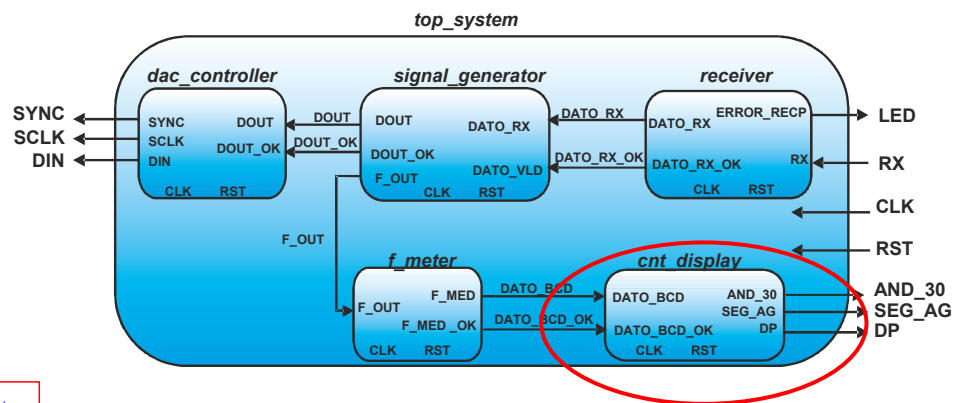
**DIN:** salida serie de los datos que se envían al DAC y que se transformarán en una señal analógica.

**AND\_30:** salidas activas a nivel bajo que se utilizan para activar los displays a través de sus ánodos. Hay que tener en cuenta que los displays son de ánodo común y los ánodos se excitan a través de transistores bipolares PNP lo que hace que los niveles de activación de los ánodos sean los contrarios: se activan con un nivel bajo.

**DP:** salida correspondiente al punto de los displays.

**SEG\_AG:** salidas correspondientes a los segmentos (a, b, ... g) de los displays de la placa.

□ Entidad *cnt\_display*.



Controla la visualización del dato de frecuencia obtenido por la entidad *f\_meter*

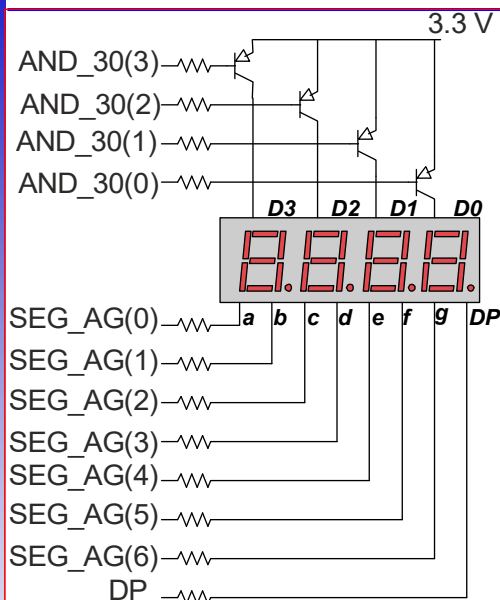
```
entity cnt_display is
  port (
    CLK      : in  std_logic;
    RST      : in  std_logic;
    DATO_BCD : in  std_logic_vector(15 downto 0);
    DATO_BCD_OK : in  std_logic;
    AND_30   : out std_logic_vector(3 downto 0);
    DP       : out std_logic;
    SEG_AG   : out std_logic_vector(6 downto 0));
end cnt_display;
```

Entrega 13 de marzo de 2023 a las 14:00

Modelado de Sistemas Computacionales (GIC)

5

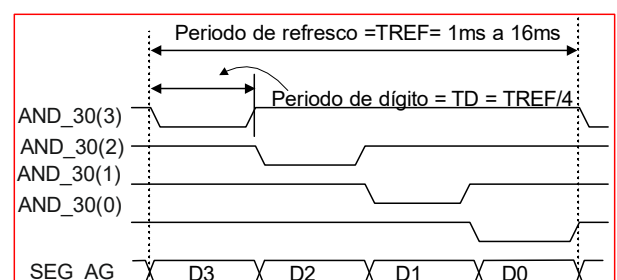
La entrega es obligatoria pero no puntuable

Entidad *cnt\_display*

El periodo de refresco (*Refresh period*), en milisegundos (ms), coincidirá con el número del puesto del laboratorio que ocupa el alumno/los alumnos. El valor que controla este factor de división se deberá asignar a una constante de nombre **CTE\_DISP**.

Para realizar las simulaciones de este apartado y con el objetivo de reducir el tiempo de simulación, se optará por el mismo valor, pero expresado en microsegundos (μs).

1. En la visualización no se deben representar los ceros a la izquierda
2. En el caso de que el valor del dato BCD sea 0000, se visualizará un 0 en el display **D0**.
3. Cuando alguno de los dígitos del puerto **DATO\_BCD** a representar en los displays no se corresponda con un valor BCD, el sistema deberá detectarlo representando la palabra FAIL.
4. Los puntos de los displays deberán permanecer siempre apagados.



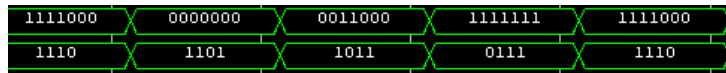
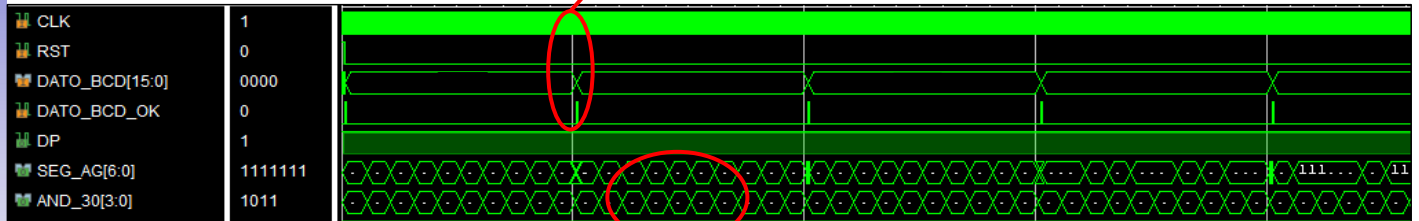
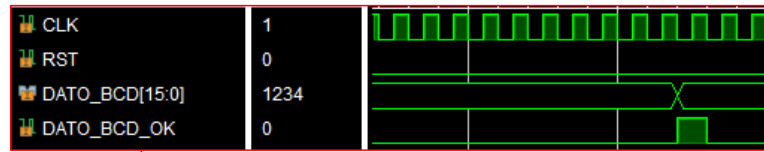
Modelado de Sistemas Computacionales (GIC)

6



## Procedimiento de diseño.

1º. Saber lo que hay que hacer



Paso 1: Almacenar.

Paso 3: Medir tiempos.

Paso 4: Multiplexar.

Paso 5: Decodificar.



## Diagrama de bloques.

Paso 1: Almacenar.

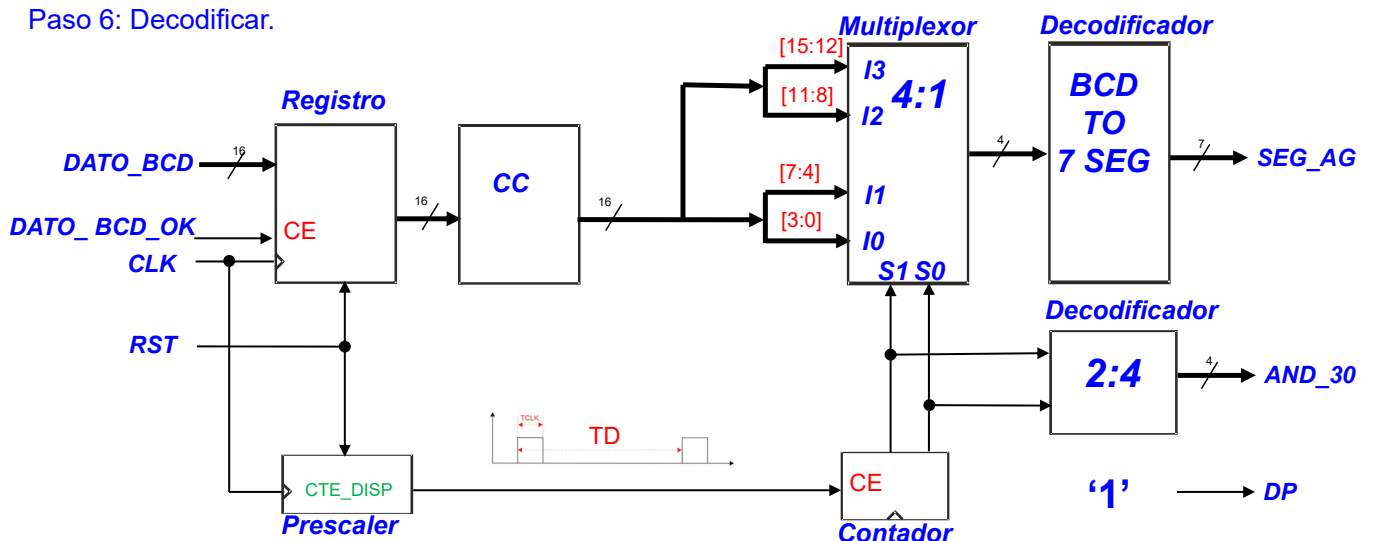
Paso 2: Verificar datos.

Paso 3: Medir tiempos.

Paso 4: Multiplexar.

Paso 5: Activar segmentos displays.

Paso 6: Decodificar.



## Entidad *cnt\_display*

### Test-bench.

```
entity cnt_display_tb is
end cnt_display_tb;

architecture sim of cnt_display_tb is
    constant CNT1 : time := 100 ns; -- Indicar valor
    signal CLK_i : std_logic := '0';
begin -- sim

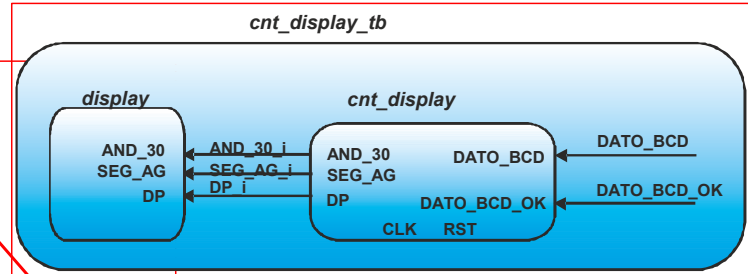
    DUT : entity work.cnt_display
        port map (
            CLK => CLK_i,
            RST => RST_i,
            DATO_BCD => DATO_BCD_i,
            DATO_BCD_OK => DATO_BCD_OK_i,
            AND_30 => AND_30_i,
            DP => DP_i,
            SEG_AG => SEG_AG_i);

    UD : entity work.display
        port map (
            SEG_AG => SEG_AG_i,
            AND_30 => AND_30_i);

    RST_i <= '1', '0' after 133 ns;
    CLK_i <= not CLK_i after 5 ns;

    process
    begin -- process

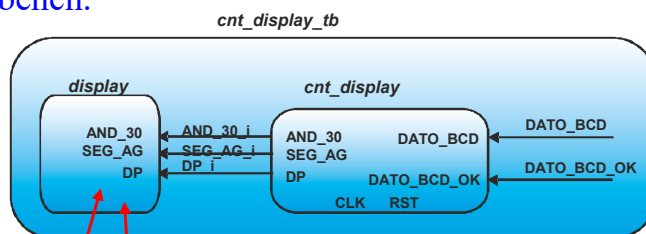
        wait for 142 ns;
        wait until CLK_i = '0';
        DATO_BCD_i <= "1000000"; -- Indicar valor
        wait for CNT1;
        . . . . .
        report "FIN CONTROLADO DE LA SIMULACION" severity failure;
    end process;
end sim;
```



¡Hay que completar!

## Entidad *cnt\_display*

### Test-bench.



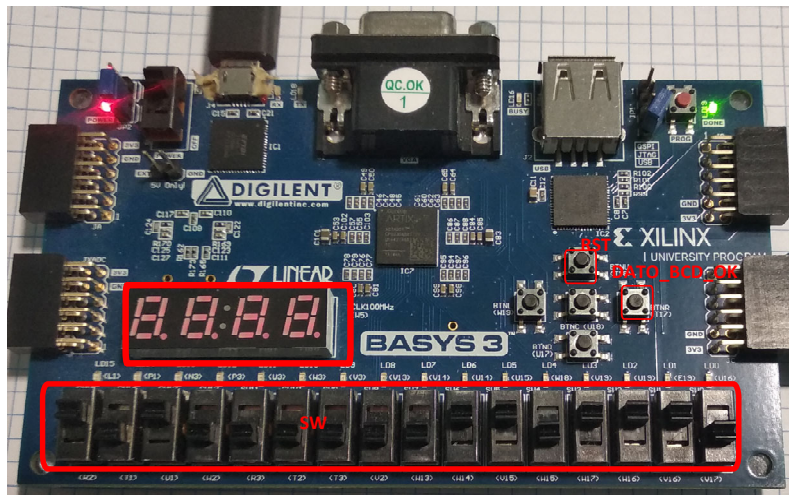
¡Este módulo se utiliza **sólo** para simulación!

¡Estas señales se deberán visualizar en las simulaciones funcional y temporal!

```
signal Data : character;
signal Displays : string(4 downto 1) := "----";

process (all) is
begin
    case SEG_AG is
        when "1000000" => Data <= '0';
        when "1111001" => Data <= '1';
        . . .
        when others => Data <= 'x';
    end case;
end process;
```

¡VHDL 2008!

☐ Verificación de la entidad *cnt\_display*.


Esta descarga es imprescindible para la evaluación del apartado. En el caso de que no funcione, se considerará el apartado como no apto.

```
# clock
set_property PACKAGE_PIN W5 [get_ports {CLK}]
set_property IOSTANDARD LVCMOS33 [get_ports {CLK}]

#RST
set_property PACKAGE_PIN T18 [get_ports {RST}]
set_property IOSTANDARD LVCMOS33 [get_ports {RST}]

#DATO_BCD_OK
set_property PACKAGE_PIN T17 [get_ports {DATO_BCD_OK}]
set_property IOSTANDARD LVCMOS33 [get_ports {DATO_BCD_OK}]

# Switches
set_property PACKAGE_PIN V17 [get_ports {DATO_BCD[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {DATO_BCD[0]}]

. . . .
```