
**GRAU D'ENGINYERIA ELECTRÒNICA INDUSTRIAL i AUTOMÀTICA**

**LABORATORIO DE ELECTRÓNICA DIGITAL**

**CURSO 21-22**

# DISEÑO DE UN RELOJ DIGITAL MULTIFUNCIÓN EN VHDL

**Profesores: Enrique Fernando Cantó Navarro**
**Esteban del Castillo**

**Alumnos: Chiara Coppola**
**Luisa Vicente Oliveira**

**Grupo de laboratorio: L5**

## Diseño de un reloj digital multifunción con VHDL

En el laboratorio se ha creado un reloj digital con VHDL que puede desempeñar las siguientes funciones:

1. *un reloj de 24h que muestra hora, minutos y segundos,*
2. *una alarma,*
3. *un cronómetro que cuenta segundos y decisegundos.*

Estas funcionalidades se pueden ver con el visor, que muestra cada operatividad según la selección que se hace mediante un multiplexor.

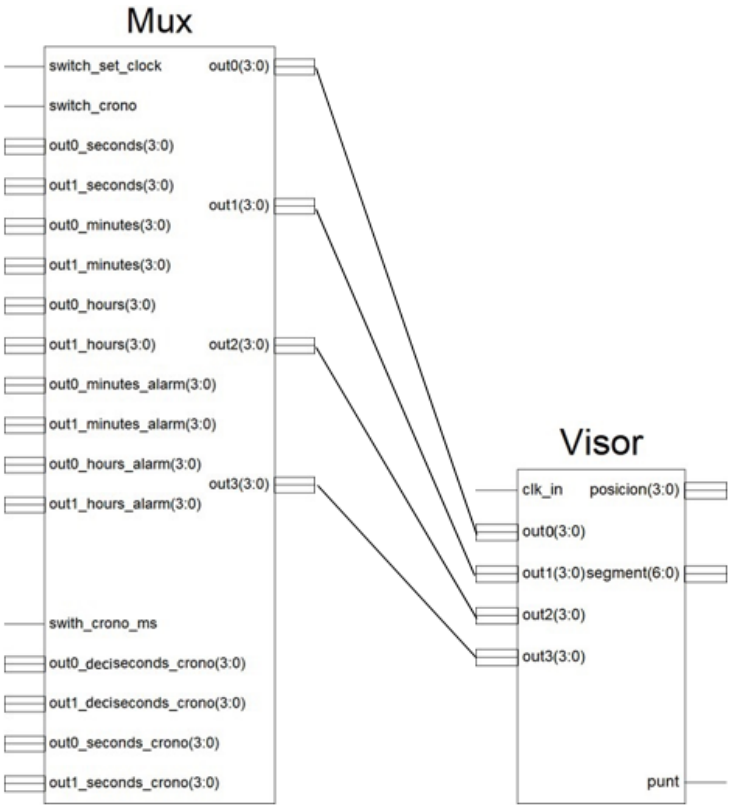


Figura 1. Multiplexor

Como se puede ver en la figura con el multiplexor se puede seleccionar, por medio de botones (*switches*) de la placa, si se quiere visualizar el reloj (horas y minutos o minutos y segundos), el cronómetro (segundos y decisegundos) o la configuración para determinar el tiempo en que se acciona la alarma.

Ejemplo de código VHDL de una salida del multiplexor:

```
out0 <= out0_seconds when switch_crono = '1'
        else out0_minutes_alarm when switch_set_clock = '1'
        else out0_deciseconds_crono when swith_crono_ms = '1'
        else out0_minutes;
```

## 1. Reloj

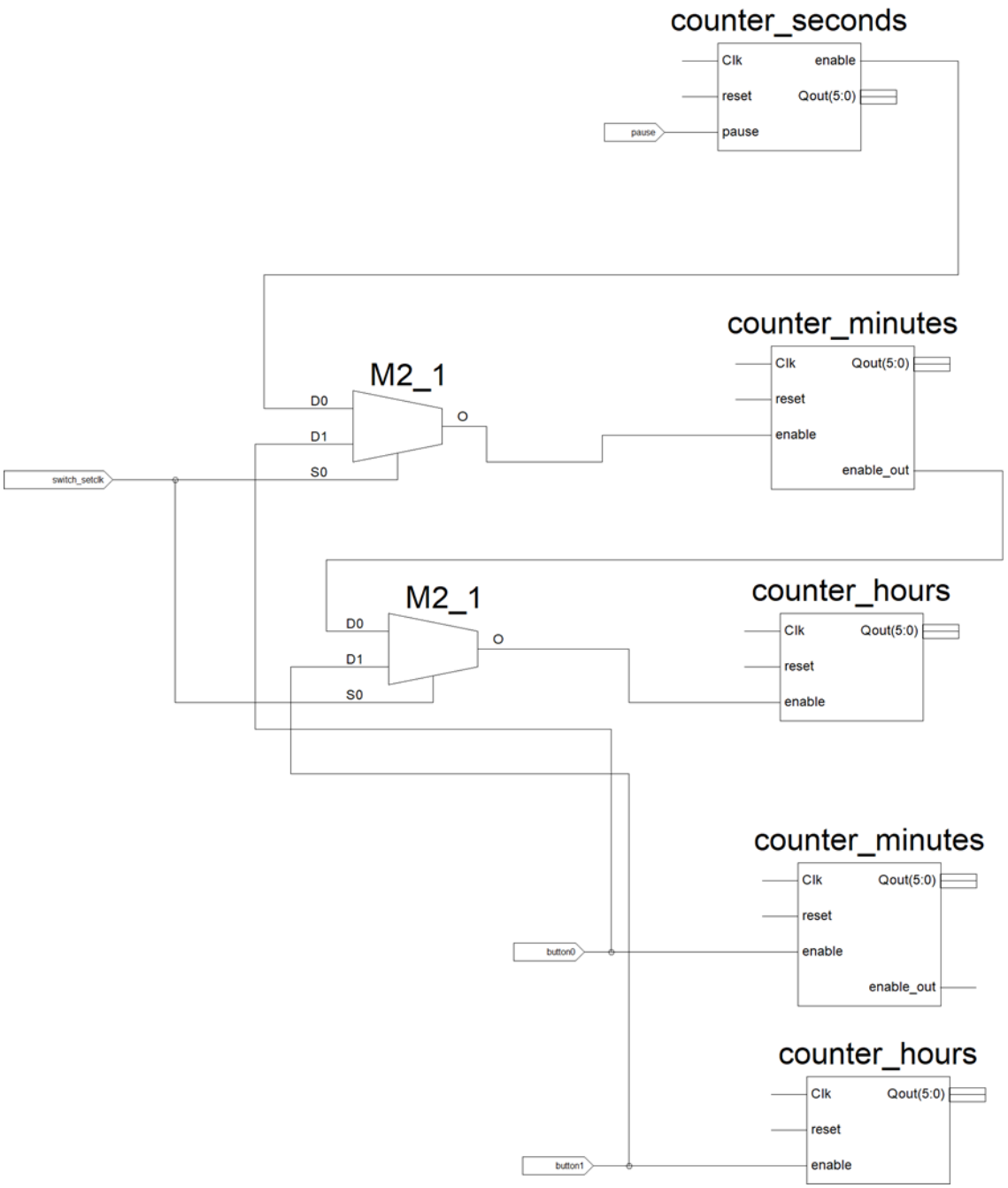


Figura 2. Esquema del contador

El reloj consta de tres contadores. El primero cuenta hasta 60 segundos y a continuación pone en marcha el segundo contador con una señal de habilitación, de esta forma, el contador de minutos sólo cuenta cuando han pasado 60 segundos. Del mismo modo, el tercer contador cuenta sólo después de haber recibido la señal de habilitación del contador de minutos, de modo que las horas se incrementan después de que hayan pasado 60 minutos y 60 segundos. En los displays, es posible mostrar el tiempo del reloj en dos formatos: HH:MM o MM:SS, según se apriete el interruptor correspondiente en la placa.

Ejemplo de código VHDL del proceso del contador de horas y minutos:
(*Se puede visualizar el uso de la señal de habilitación*)

```
process(Clk, reset, enable)
    begin
                if (reset='1') then
                      actual <= "000000";
                elsif (Clk'event and Clk='1' and enable = '1') then
                      actual <= seguent;
                end if;
    end process;
```

Además, como se muestra en la figura 2, con la ayuda de los multiplexores 2:1, es posible elegir entre mostrar la hora 00:00 o inicializar el reloj con una hora elegida mediante botones - para esta última parte también se han utilizado contadores.

---

**2. Alarma**

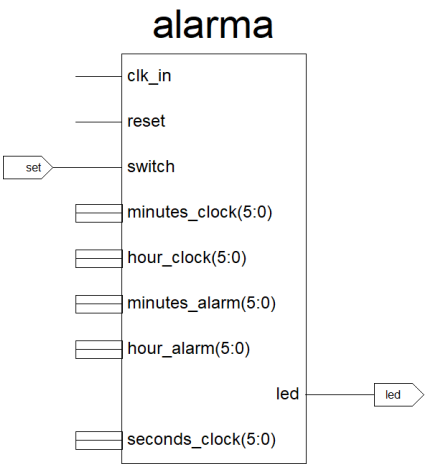


Figura 3. Alarma

Levantando el interruptor correspondiente en la placa se entra en el modo de configuración de la alarma. En la placa, con los dos primeros botones empezando por la derecha debajo de los displays es posible elegir las horas y los minutos para los que se va a configurar una alarma. Bajando el interruptor se sale del modo de configuración.

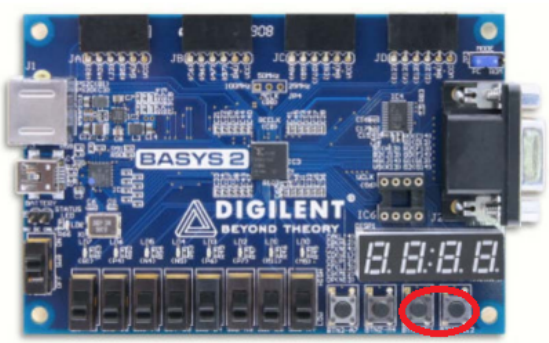


Figura 4. Botones para configurar la hora en que “suena” la alarma

Cuando el reloj alcanza la hora y los minutos seleccionados en el modo de configuración, un led se encenderá y permanecerá encendido hasta que se apague con un interruptor. Para ello, se utiliza una máquina de estado con sólo dos estados. Cuando el recuento del tiempo del reloj es igual al tiempo para el que se ha establecido una alarma, el estado pasa de cero a uno y el LED se enciende. Además del control de la hora y los minutos, existe un control de los segundos, de modo que la máquina de estado cambia su estado de 0 a 1 sólo una vez: cuando acaba de pasar la hora de alarma seleccionada. Sin esta condición, la máquina de estado pasaría constantemente del estado cero al estado uno provocando el parpadeo del led.

Código VHDL de la alarma:

```
cmp <= '1' when (minutes_clock = minutes_alarm) and (hour_clock = hour_alarm) and
(seconds_clock = "000000")
                    else '0';
        led <= '1' when state = '1'
                    else '0';

        process(reset,clk_in) is
                begin
                    if (reset = '1') then
                            state<= '0';
                    elsif (clk_in'event and clk_in = '1') then
                            if (state = '0' and cmp = '1' )then
                                    state<= '1';
                            elsif (state='1' and switch = '1') then
                                    state<='0';
                            end if;
                 end if;
        end process;
```

---

### 3. Cronómetro de segundos y decisegundos

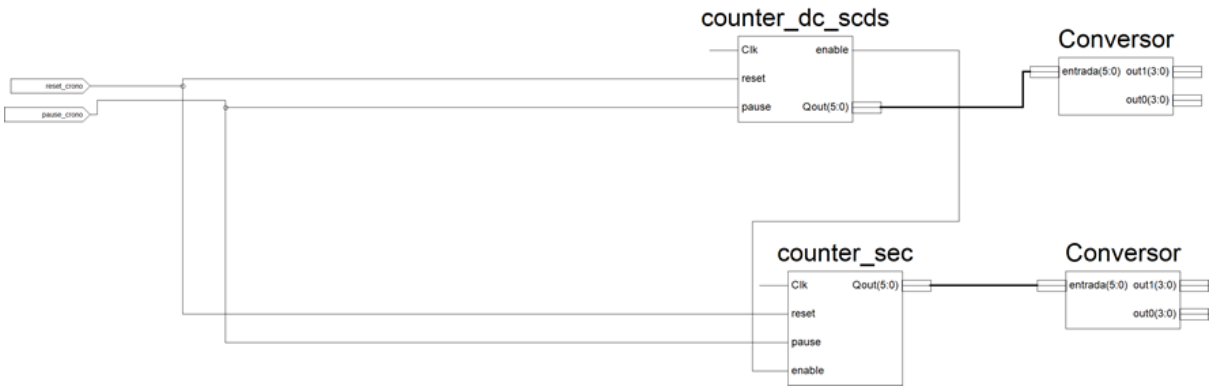


Figura 5. Esquema del cronómetro

Mediante otro interruptor también es posible visualizar el cronómetro de segundos y decisegundos que se construye utilizando el mismo contador de segundos que del reloj, pero cambiando la frecuencia a la que se conecta el componente. De modo que el contador

de segundos solo empieza a contar cuando el contador de decisegundos ha llegado a los diez segundos – siguiendo la misma lógica usada en el reloj, para el cronómetro también se ha utilizado una señal de habilitación.
Es importante observar que el reinicio y la pausa de este cronómetro es independiente del reinicio y la pausa del reloj.

Código VHDL del contador de decisegundos:

```
seguent <= "000000" when actual= "001010" else actual+1;
enable <= '1' when actual= "001010" else '0';
    process(Clk, reset)
    begin
                if (reset='1') then
                      actual <= "000000";
                elsif (Clk'event and Clk='1' and pause='0') then
                      actual <= seguent;
                end if;
    end process;
```

---

**Plantilla UCF**

A continuación se muestra una parte de la plantilla UCF utilizada para el diseño de este reloj multifuncional. En concreto, es la lista de correspondencias entre las señales de salida y los interruptores, leds y botones de la placa:

```
NET "button0" LOC = G12;
NET "button1" LOC = C11;

NET "switch_set" LOC = L3;
NET "switch_crono" LOC = P11;
NET "set" LOC = K3;

NET "led" LOC = G1;

NET "switch_crono_ms" LOC = B4;
NET "switch_setclk" LOC = G3;

NET "reset_crono" LOC = A7;
NET "pause_crono" LOC = M4;
```