



UNIVERSIDAD  
DE  
CÓRDOBA

## Proyecto Personal: Despertador

Computación de Altas Prestaciones

**Carlos Checa Moreno**

26507320G

i02chmoc@uco.es

Curso Académico 24/25



# ÍNDICE GENERAL

---

<b>CAPÍTULO 1</b>	<b>INTRODUCCIÓN .....</b>	<b>2</b>
<b>CAPÍTULO 2</b>	<b>DISEÑO .....</b>	<b>3</b>
2.1	RELOJ .....	3
2.2	ALARMA.....	4

# Capítulo 1

## INTRODUCCIÓN

---

En esta memoria se llevará a cabo una explicación del proyecto final de la asignatura Computación de Altas Prestaciones.

En este trabajo se han usado los conocimientos adquiridos en las prácticas realizadas durante el transcurso de las sesiones de la asignatura. En estas prácticas se ha estudiado el comportamiento de la FPGA BASYS 3 - Artix-7 XC7A35T. Se ha usado el programa Vivado 2024.2, en el cual se ha usado el lenguaje de programación VHDL para programar diferentes comportamientos en nuestra placa.

Se ha llevado a cabo el desarrollo del despertador mediante el uso de Display, LEDs, un switch y botones.

# Capítulo 2

## DISEÑO

---

### 2.1 Reloj

Para empezar, desarrollé el reloj. Para esta tarea tenemos un proceso secuencial que opera a cada flanco de subida del reloj. Dentro de este proceso, si no se está en modo de configuración de alarma (`alarm_mode = '0'`), se comienza a incrementar un contador llamado `contador_segundos`. Este contador actúa como un divisor de frecuencia y su función es detectar el paso de un segundo real. Como tenemos un reloj de 100 MHz, cada vez que el contador llega a 100 millones de ciclos habrá transcurrido un segundo. En ese momento, se reinicia el contador y se incrementa la variable `segundos`.

Si los segundos llegan a 59, se resetean a 0 y se incrementa el contador de minutos. A su vez, si los minutos alcanzan los 59, se reinician también y se incrementa el contador de horas. Finalmente, si las horas alcanzan el valor máximo de 23, el ciclo se reinicia a las 00:00:00. De esta forma, se implementa un reloj digital funcional.

También hay que tener en cuenta que al igual que con la práctica de la palabra en el Display, en este caso también hay que usar multiplexación para mostrar los 4 números. Uso una señal llamada `refresco` que cambia de valor cada cierto número de ciclos de reloj (en este caso, cada 50.000 ciclos), permitiendo que en cada iteración del proceso de multiplexado se actualice un solo dígito del display. Dependiendo del valor de `refresco`, se selecciona un dígito específico y se carga el patrón correspondiente en `SSEG_CA`, mientras que las líneas `SSEG_AN` activan únicamente el dígito en cuestión.

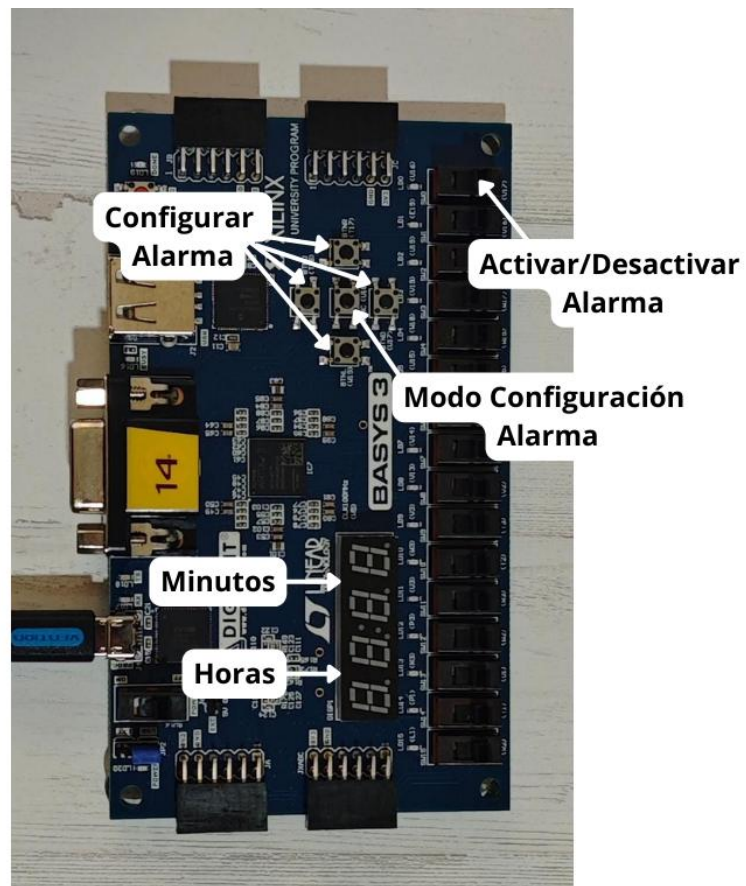
## 2.2 Alarma

El cambio entre el modo de reloj normal y el modo de configuración de alarma se realiza al presionar el botón RST. Cuando se entra al modo de alarma, el sistema guarda la hora actual en las variables `backup_horas` y `backup_minutos`, y habilita el ajuste manual de la hora, no para el reloj en sí, sino para definir a qué hora debe sonar la alarma. Cuando se sale del modo de configuración, la hora ingresada se guarda como la hora de la alarma (`alarm_horas` y `alarm_minutos`) y se restauran las variables de hora originales del reloj desde el respaldo, asegurando que la configuración no afecte el tiempo real que llevaba el reloj.

Durante el modo de configuración, los botones `BTN(0)` a `BTN(3)` permiten ajustar por separado las unidades y decenas de minutos y horas.

Una vez que se ha establecido la hora de la alarma y se ha vuelto al modo normal, el sistema compara continuamente la hora actual del reloj con la hora programada para la alarma. Si coinciden y el interruptor `SW` está activado, la señal `alarma_activa` se pone en alto. Esto provoca que todos los LEDs del sistema se enciendan como indicación visual de que la alarma ha sido activada. En caso de que el interruptor `SW` esté apagado o la hora ya no coincida, la alarma se desactiva automáticamente .

---



**Figura 1.** Diseño Alarma FPGA