

CIRCUITOS SECUENCIALES

Objetivos

- El alumno aprenderá sobre el funcionamiento de circuitos secuenciales mediante la instanciación de flip-flops, para realizar contadores sencillos de diferentes módulos. Para esto, se deberá diseñar e implementar un reloj digital en formato de 12 horas con alarma programable.
- También, se hará uso de displays de 7 segmentos para el despliegue de información en la tarjeta de desarrollo, por lo que el alumno implementará una máquina de estados simple para la codificación de BCD a 7 segmentos.
- Finalmente, el alumno hará uso de un módulo IP Core del ambiente de desarrollo para implementar un divisor de frecuencia que servirá para obtener una frecuencia derivada de la señal de reloj de la tarjeta de desarrollo.

Actividades

1. Implementa la lógica secuencial del reloj digital utilizando contadores MOD10 y MOD6 integrados con flip-flops, para contar segundos y minutos.
2. Implementa la lógica secuencial del reloj digital utilizando contadores MOD10 y MOD2 integrados con flip-flops, para contar horas en formato 12. Incluye un indicador para AM y PM (ya sea un LED o un punto en el display).
3. Utiliza el codificador de BCD a 7 segmentos para desplegar las horas y minutos en cuatro displays de 7 segmentos (incluido en la tarjeta de desarrollo). Consulta el [Manual de Referencia de la tarjeta NEXYS 4 DDR](#) (sección 10.1) para verificar el pin-out y la configuración correspondiente al display.
 - a. La tarjeta de desarrollo cuenta con un cristal de cuarzo para generar una señal de reloj de 100 MHz. Para implementar el decodificador, será necesario implementar un divisor de frecuencia para la señal de reloj de 100 MHz a 100 Hz. Para generar este divisor, utiliza el Clocking Wizard del ambiente de desarrollo.
4. Utiliza un LED que parpadee la frecuencia del segundero.

Actividades adicionales

5. Implementa la alarma programable. Para esto, toma en cuenta las siguientes restricciones:
 - a) Utilizar un push-button (PB1) para mostrar la alarma programada o la hora.
 - Mientras no se presione PB1, se mostrará la hora con normalidad
 - Al presionar PB1, se mostrará la alarma programada.
 - b) Utilizar otro PB (PB2) y 8 slide-switches (SW) para programar la alarma.
 - Para esto deberá mantener presionado PB2 y al mover los primeros 4 SWs, se programarán las decenas de horas: **10:25**. Las unidades de horas serán programadas con los 4 SWs restantes: **10:25**.
 - Si PB2 no se presiona, por defecto modificará las decenas de minutos con los primeros 4 SW: **10:25**, y las unidades de minutos con los 4 SW restantes:

10:25. Debido a esta restricción será necesario programar primero las horas y luego los minutos.

- c) Utilizar un LED (LED2) para indicar que la alarma está activa.
 - LED2 se mantendrá encendido durante 1 minuto (60 ciclos de Segundos_unidades) y se apagará pasando este tiempo.
 - Utilizar otro botón (PB3) para desactivar la alarma al presionarlo una vez, si no ha pasado el minuto que se mantendrá la alarma activa
- d) Utilizar un LED (LED3) para indicar que la función de alarma está activada
 - LED3 se mantendrá encendido mientras que el usuario desee que la función de alarma esté activada, y apagado cuando el usuario deshabilite dicha función.
- e) Utilizar un botón (PB4) para activar o desactivar la función de alarma.
 - Si el usuario presiona una vez PB4, se encenderá automáticamente LED3, indicando que la función de alarma está habilitada
 - Al deshabilitar la función de alarma, a pesar de que la hora coincida con la alarma programada, esta será ignorada y no encenderá LED2.

Reporte

En tu reporte técnico, incluye lo siguiente:

1. Cabecera (Nombre del curso, número de práctica, fecha de entrega, nombre y matrícula de alumnos)
2. Enlace a GitHub de los códigos en VHDL
 - a. Flip-flop J-K
 - b. Contadores MOD10, MOD6 y MOD2
 - c. Reloj integrado
3. Enlace a YouTube de la demostración de implementación en FPGA
4. Diagrama esquemático del reloj
5. Evidencias de simulación con capturas de pantalla de línea de tiempo
6. Conclusiones