**Objetivo**

El alumno aprenderá la manera de organizar un proyecto de manera modular y separarlo en diferentes archivos, con la finalidad de que vaya construyendo su propia biblioteca de módulos funcionales, y pueda reutilizar los módulos generados en otros proyectos.

**Introducción**

Para esta práctica, igual que las dos anteriores, deberemos recordar conocimiento visto en la materia de Diseño Digital Moderno. Además de eso, como se va a hacer uso de un servomotor para la realización de este proyecto, se requiere el conocimiento en el funcionamiento de éstos.

Los conceptos básicos más importantes en esta práctica son: entendimiento de VHDL, incluyendo la sintaxis básica, la estructura de los procesos, las señales y las entidades. Como se nos dio el código para hacer uso de un servomotor en el primer punto, deberemos estar familiarizado con su diseño y la forma en que se controla. Debemos conocer cómo crear y utilizar módulos genéricos en VHDL, los cuales nos permiten parametrizar diseños y reutilizarlos con diferentes configuraciones.

Los servomotores se controlan mediante pulsos PWM (Modulación por Ancho de Pulso), por lo que se deberá encontrar la manera que éstos se muevan con el uso de los interruptores de presión tipo push button. Para esto se deberá tener conocimiento electrónico de cómo se conectan a una FPGA.

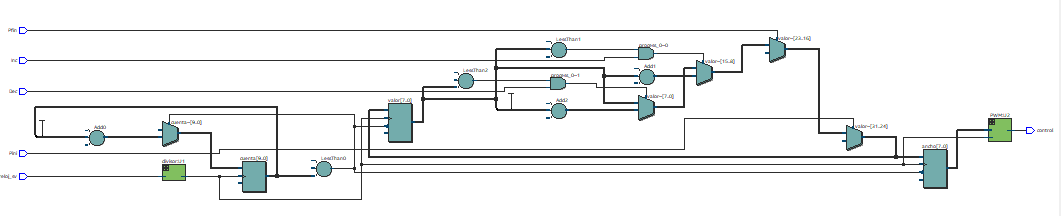
Debemos tener una idea clara del diseño del sistema, incluyendo cómo se conectarán los interruptores de presión a la FPGA y cómo se generará la señal de control para los servomotores. Para esto se tendrá que analizar el código proporcionado por el profesor y realizar las modificaciones pertinentes para lograr lo planeado si se desea que los servomotores se controlen de forma independiente.

Para esta práctica se realizará un video en el cual se podrá ver el funcionamiento de los servomotores de forma independiente mediante los push buttons, el código se ejecutará, compilará y meterá en la tarjeta usando el software de Quartus.

**Desarrollo**

Utilizando el módulo genérico para controlar un servomotor diseñado en el primer punto, construir un sistema que haga que dos servomotores se mueven independientemente, cada uno de estos controlado por dos interruptores de presión tipo push button que al presionarlo harán avanzar o retroceder el eje del motor.

Primeramente, se muestra el RTL del diseño físico para su fácil comprensión:



En este caso se realizó lo siguiente:

Primeramente, se escribieron los tres códigos correspondientes, cada código representa un módulo distinto:

**Explicación del módulo divisor de frecuencia:**

Dentro de la arquitectura, hay un proceso que se activa cuando hay una señal de subida en la señal de reloj. El proceso contiene las siguientes partes:

* **Constante N**: Se declara una constante llamada "N" con un valor de 11. Esta constante está relacionada con el tamaño del vector cuenta.
* **Variable "cuenta"**: Se declara una variable "cuenta" como un vector de 28 bits inicializado en (X="0000000"). Esta variable se utiliza para llevar la cuenta la subida del reloj.
* **Condición if**: Dentro del proceso, se verifica si ha ocurrido una subida en la señal de "reloj" utilizando la función "rising\_edge". Si es así, se incrementa la variable "cuenta" en uno.
* **Asignación de salida**: Al final del proceso, se asigna el valor de la posición N del vector "cuenta" a la señal de salida "div\_reloj". Esto divide la salida en 11, lo que provocara que sirva para la posición del servomotor posteriormente.

**Explicación del módulo PWM:**

Proceso: Dentro de la arquitectura, hay un proceso que se activa cuando hay un cambio en la señal de "reloj\_pwm" El proceso contiene las siguientes partes:

Variable "cuenta": Se declara una variable llamada cuenta como un número entero con un rango de 0 a 255, inicializada en 0. Esta variable se utiliza como un contador que se incrementa con cada subida de "reloj\_pwm".

Condición if: Dentro del proceso, se verifica si el valor de "reloj\_pwm" es '1' y ha ocurrido un evento de flanco de subida. Si ambas condiciones son verdaderas, se ejecuta el código dentro del bloque "if".

Incremento del contador y generación de PWM: Dentro del bloque if, se incrementa la variable "cuenta" y luego se aplica una operación "mod 256" para asegurarse de que el contador esté en el rango de 0 a 255. Luego, se compara el valor de "cuenta" con el valor de entrada "D". Si "cuenta" es menor que "D", la salida "S" se establece en '1', de lo contrario, se establece en '0'.

**Explicación del módulo principal:**

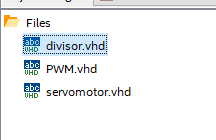
El código principal del módulo está dentro de un proceso que se activa cuando hay un cambio en la señal "reloj\_serv" o cuando cualquiera de las señales de entrada "Pini," "Pfin," "Inc," o "Dec" cambian. El proceso contiene las siguientes partes:

Se declaran dos variables: "valor" como un vector de 8 bits y "cuenta" como un número entero en el rango de 0 a 1023.

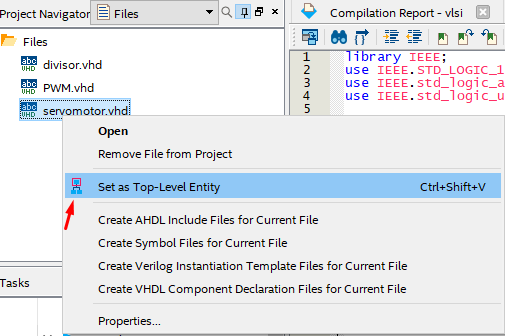
Dentro del proceso, se verifica si la señal "reloj\_serv" es '1' y ha ocurrido un cambio de subida. Si ambas condiciones son verdaderas, se ejecuta el código dentro del bloque "if".

En función del valor de "cuenta," las señales de entrada ("Pini," "Pfin," "Inc," y "Dec") y otras condiciones, se ajusta el valor de "valor." Luego, se reinicia "cuenta" a 1023 y se asigna el valor de "valor" a la señal "ancho," que controla la señal PWM de salida "control."

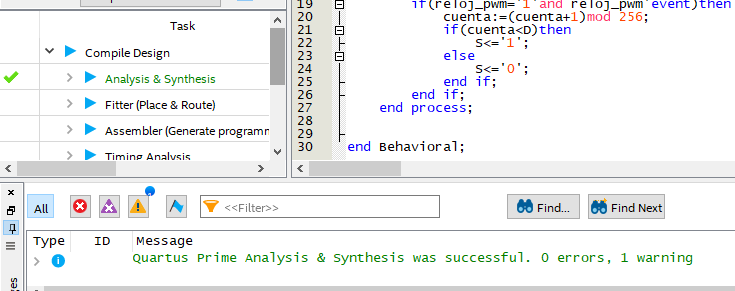
Una vez explicados los módulos lo que hicimos fue poner los tres archivos en una misma jerarquía como se ve en la imagen:



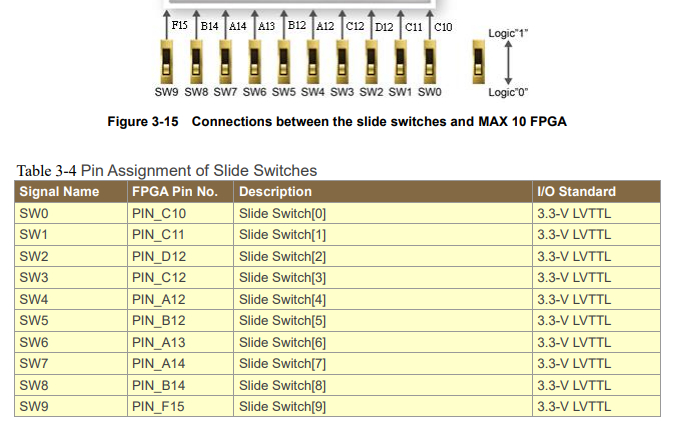
Posteriormente le daremos el privilegio de TOP al archivo servomotor.vhd que es el módulo principal, que lo realizaremos de la siguiente manera:



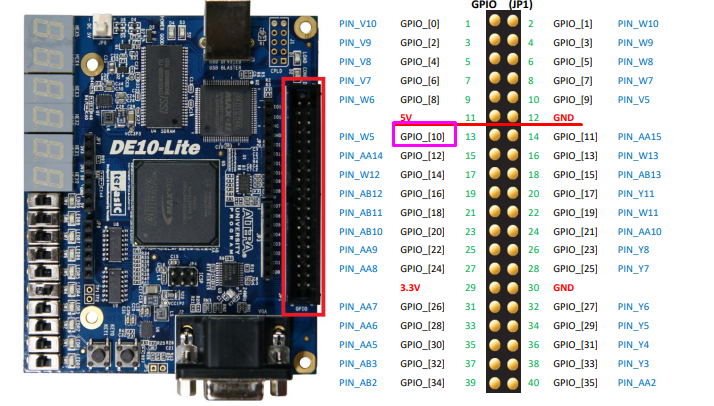
Una vez realizado esto deberemos ejecutar el programa para verificar que funcione de la manera correcta:



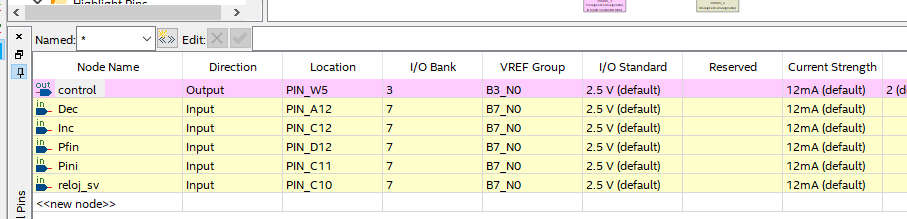
Una vez comprobado pasaremos al pin planner para asignar los pines correspondientes, para esto nos guiaremos del manual de la de10lite, como solo teníamos dos botones decidimos utilizar los switchs, ya que contamos con 4.



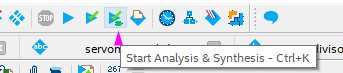
Y para la salida utilizamos el módulo de salida, además de la tierra y voltaje de 5V correspondientes de la placa.



Por lo que en el pin planner nos queda de la siguiente manera:

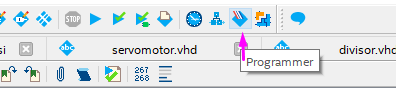


Para revisar que este de la manera correcta una vez más, comenzamos el análisis y la síntesis:

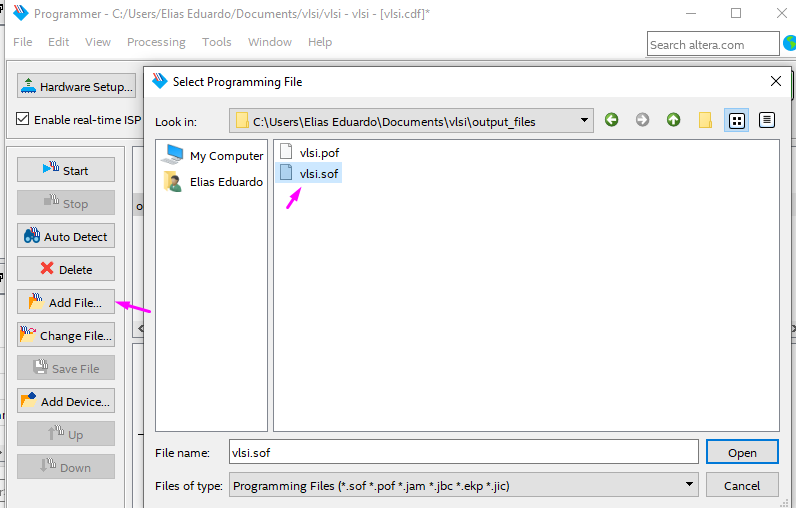


Una vez listo este paso el siguiente paso es subir el programa a la placa y conectar el servomotor, para subir la placa debemos realizar lo siguiente:

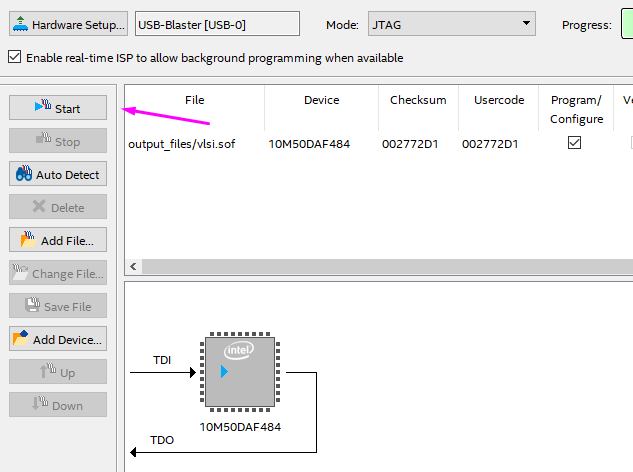
Primero debemos de conectar la placa a la PC, para eso debemos presionar programmer:



Dentro del programmer debemos añadir un nuevo archivo agregaremos el .SOF



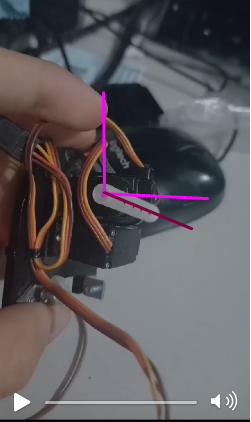
Y casi para finalizar le daremos en start:



Por lo que teniendo como resultado la salida del servomotor que nos debe entregar un movimiento a partir del contador respecto al ciclo de reloj tenemos el siguiente movimiento:

**Resultados**

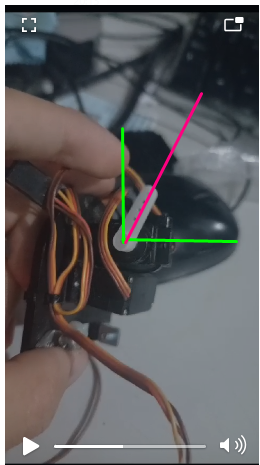
Al iniciar el servomotor comienza en la posición 0 y conforme va aumentando el tiempo aumenta o decrementa según sea la salida de la tarjeta como se puede ver en las siguientes imágenes:



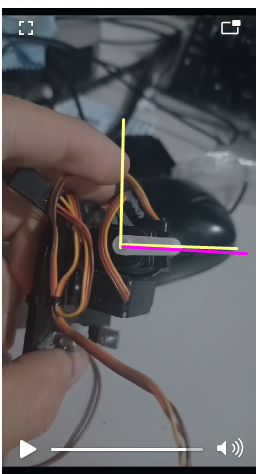


En la siguiente imagen se puede ver como transcurridos unos segundos de que inicio el video, la posicion inicial del servomotor ha cambiado aumentando en este caso.





En la siguiente imagen una vez cambiado el estado de la tarjeta podemos ver como el contador ahora disminuye y el ciclo del servomotor retrocede, provocando que cambie el sentido de giro.



**Link del video:** <https://youtube.com/shorts/6g9NwBdopD8?feature=share>

**Bibliografía**

* <https://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=China&No=1021&FID=a13a2782811152b477e60203d34b1baa> Consultado el 05/09/23