

TRABAJO DE SISTEMAS ELÉCTRONICOS DIGITALES  
Curso 2021-2022

# ASCENSOR CONTROLADO POR UNA FSM



López de Meneses Calvo, Laura 54952

Ortega Monge, María 54777

Villajos Mateo, Laura 54904

# ÍNDICE:

1. DESCRIPCIÓN DEL TRABAJO.
  - 1.1 ANÁLISIS DE REQUISITOS.
  - 1.2 DIVISIÓN FUNCIONAL DEL DISEÑO.
  - 1.3 ARQUITECTURA DEL SISTEMA.
  - 1.4 VERIFICACIÓN FUNCIONAL.
  - 1.5 CIERRE DE PRE-SÍNTESIS.
  - 1.6 SÍNTESIS.
  - 1.7 PROCESO DE VERIFICACIÓN POST-SÍNTESIS.
  - 1.8 CIERRE DEL DISEÑO.
2. PROBLEMAS OCASIONADOS Y SOLUCIONES ADOPTADAS.
3. CONCLUSIÓN.
4. ENLACES DE GITHUB Y AL VIDEO DEMOSTRATIVO.

# 1. Descripción del trabajo

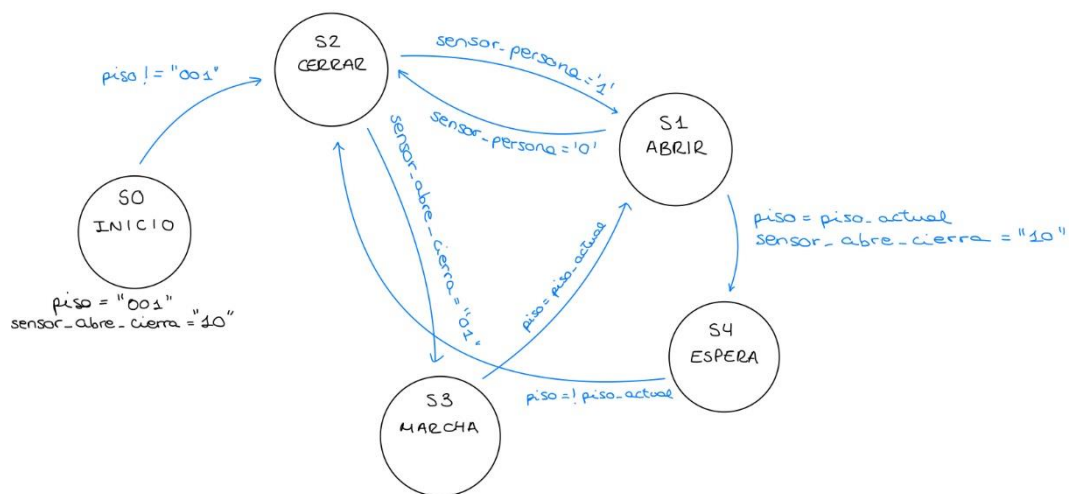
Teniendo como base, la información que nos han impartido este curso en la asignatura de Sistemas Electrónicos Digitales, vamos a realizar la descripción hardware en VHDL del funcionamiento de un ascensor.

El proceso de diseño contara con todas las fases necesarias.

## 1.1 Análisis de requisitos.

Para nuestro diseño queremos implementar un ascensor con un número determinado de pisos, en este caso 7, que cuando entre una persona espere a que se presione un botón, una vez pulsado ese botón será nuestro destino y a menos que entre otra persona (lo que activaría el sensor de presencia y se volverían a abrir las puertas), se cerrarán las puertas y subirá o bajará los pisos determinados hasta llegar al destino, finalmente se abrirán las puertas para que las personas puedan bajarse y se quedará en reposo en ese piso.

Durante ese proceso de subida y bajada estará activa la salida motora en condiciones de subida y bajada.



## 1.2 División funcional del diseño.

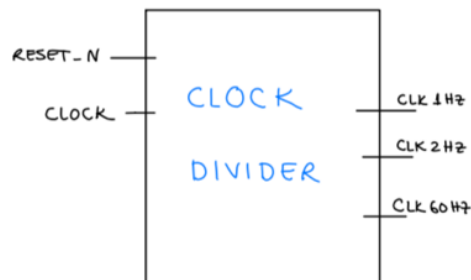
Desarrollo a un nivel no trivial

Es necesario dividir el diseño total como pequeñas funcionalidades más básicas, de tal forma que abordar la codificación de cada elemento se remonte a la funcionalidad más simple posible.

### Clock divider.

Es el módulo encargado de adaptar la frecuencia de la placa a cada uno de los módulos.

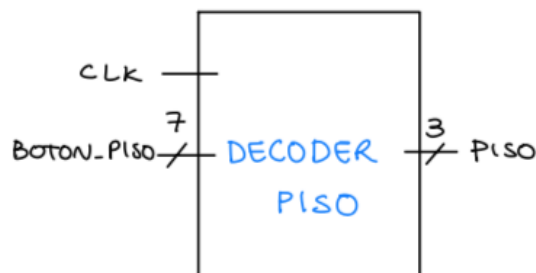
Su entrada es el reloj de la propia placa FPGA y genera las distintas señales que le llegan a cada bloque.



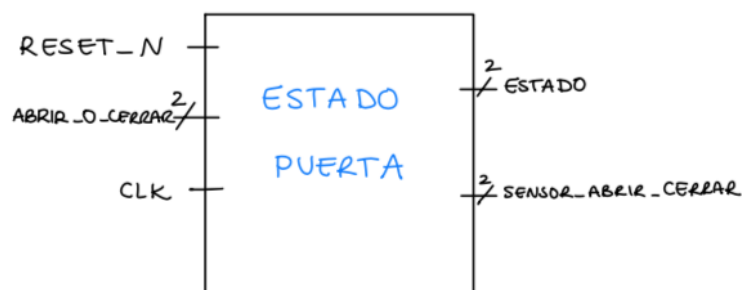
### Decoder\_piso.

Este bloque se utiliza para recibir 7 bits de información (uno por piso) y obtener 3 bits de información a la salida.

Este bloque lo utilizamos tanto a la salida de Estado\_piso como en la entrada de la FSM.



### Estado\_puerta.



Este bloque simula el comportamiento de la puerta, recibe la señal **abrir\_o\_cerrar** del módulo control puerta, que proporciona al módulo la información sobre el

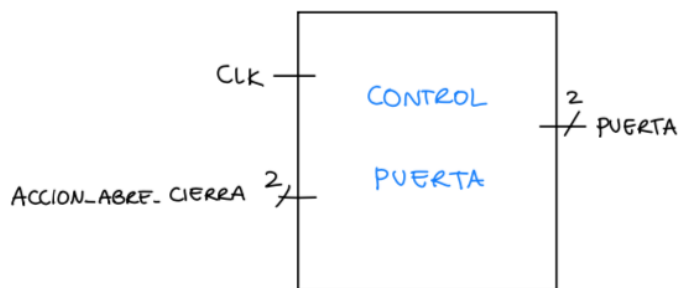
movimiento de los motores de las puertas. La salida **sensor\_abrir\_cerrar** es la que posteriormente va a la FSM que coordina ambos bloques.

### Control\_puerta.

Este bloque se encarga de abrir y cerrar las puertas del ascensor cuando es preciso, en concreto de activar el motor de puerta.

A este bloque le llega la señal **acción\_abre\_cierra** de la FSM que le indica si debe abrir o cerrar la puerta. Tiene como salida **puerta** que a su vez está conectada a estado puerta.

Además, tiene como entradas un **reset** para resetear el módulo y una señal de reloj.

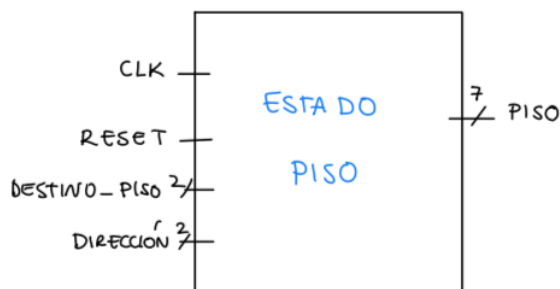


### Estado\_piso.

Este bloque simula el comportamiento del ascensor.

A este bloque le llega la entrada **dirección** que le indica si el ascensor está subiendo o bajando. Tiene como salida **piso** de 7 bits que llegará hasta el decodificador.

Además, tiene como entradas un **reset** para reiniciar el módulo y una señal de reloj.

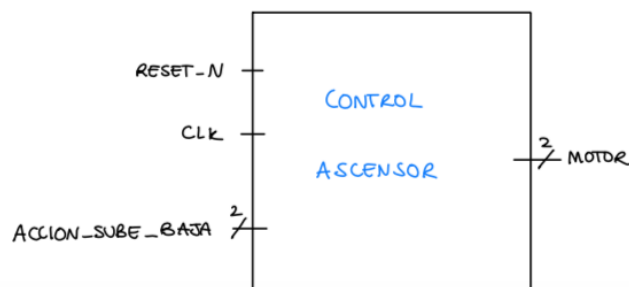


### Control\_ascensor.

Este módulo controla la subida y la bajada del ascensor, lo que sería el motor del ascensor.

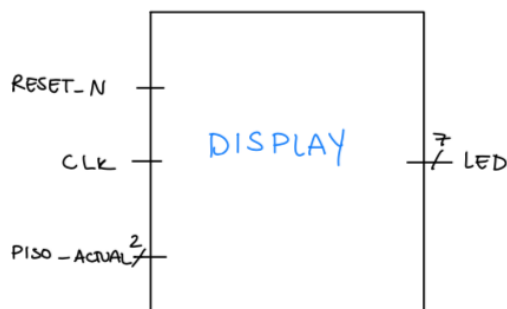
Tiene como entrada una señal de reloj y de **reset**. Además, cuenta con la entrada **accion\_sube\_baja** que es la señal proveniente de la FSM.

Como salida tiene la señal **motor** que le llegará a estado\_piso como **dirección** y es el sentido que lleva el ascensor (subiendo o bajando).



### Display.

Se trata de un display de 7 segmentos que muestra el piso actual del ascensor, recibe la señal piso actual y la muestra.

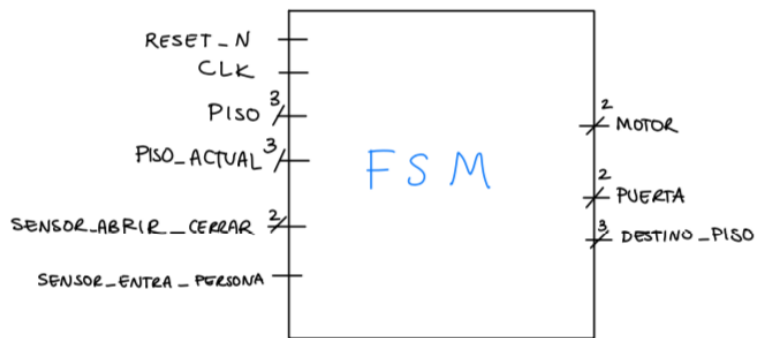


### FSM.

Este es el bloque encargado del funcionamiento del conjunto total de los módulos que se acaban de definir.

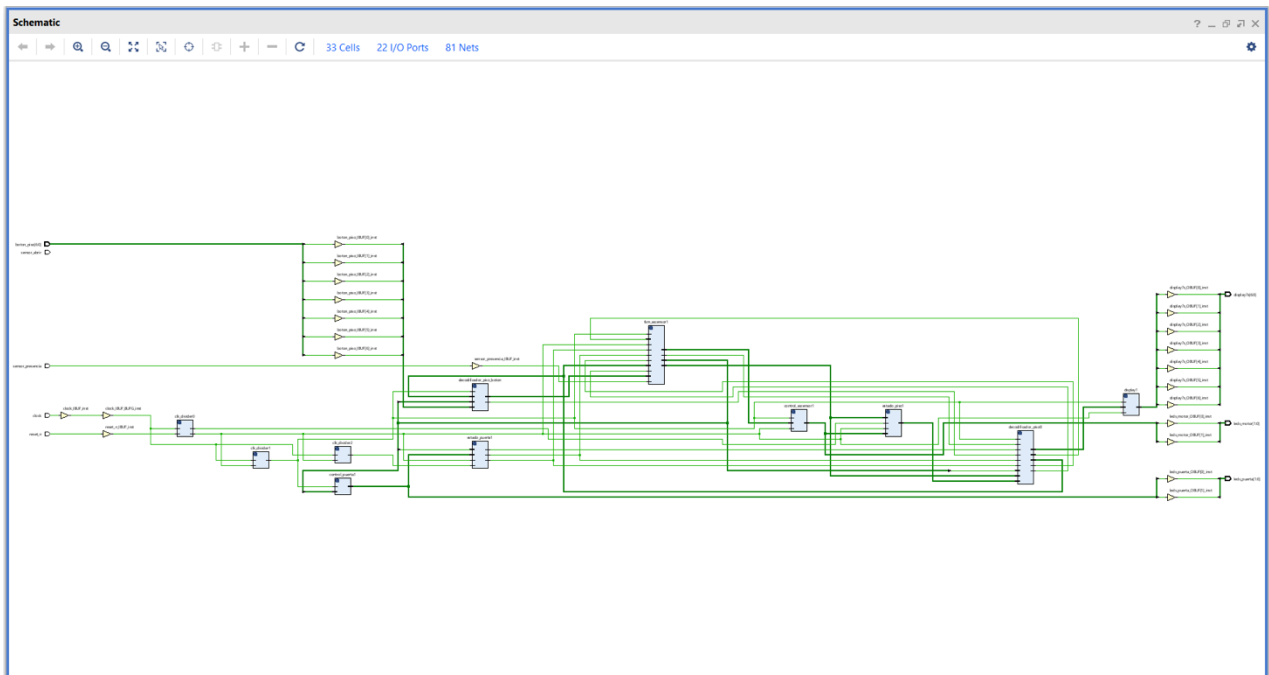
Le llegan las señales de **reset**, **clock**, **sensor\_abrir\_cerrar**, **piso\_actual**, **piso**.

Y sus salidas son: **puerta** y **motor**.

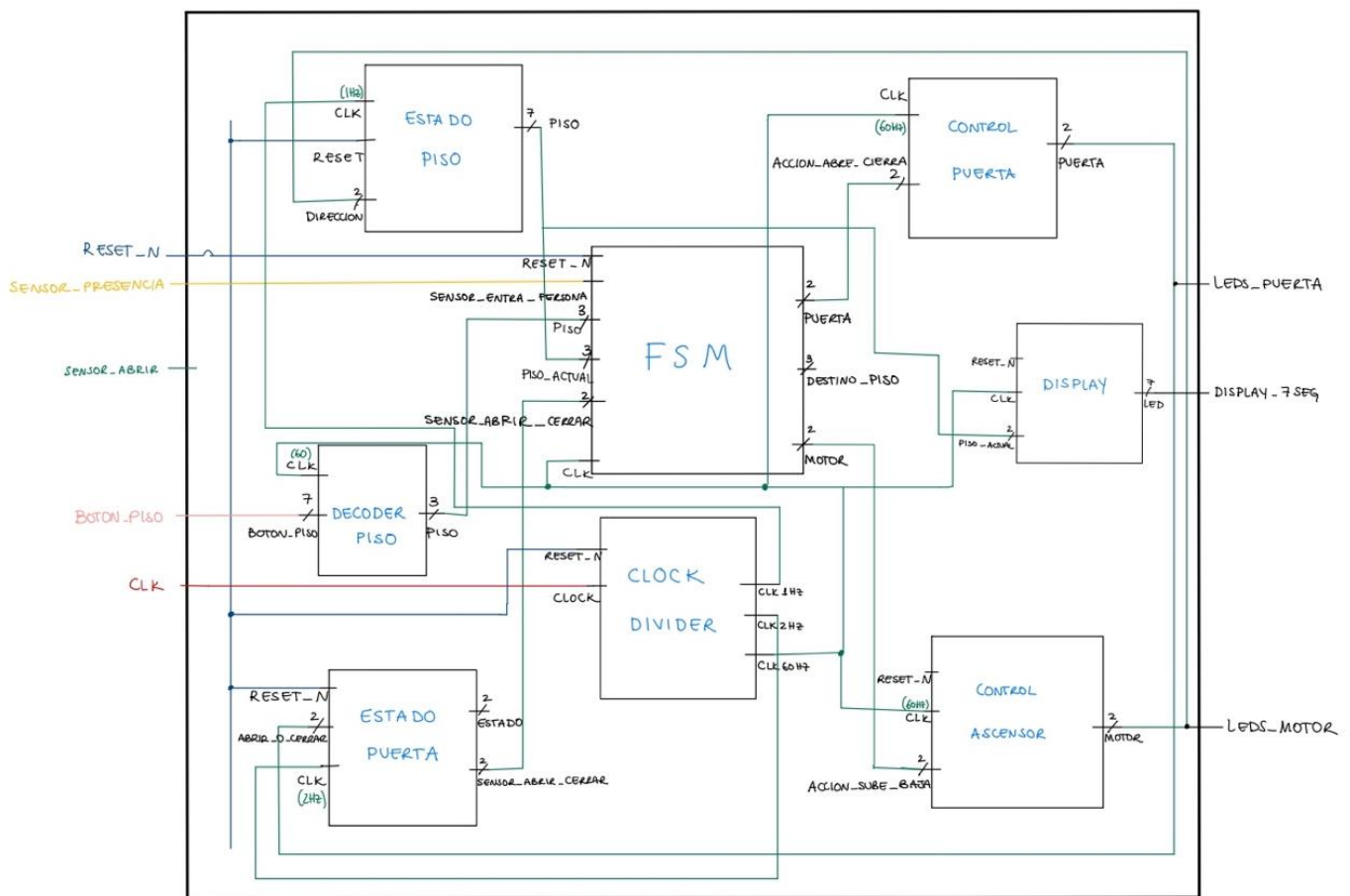


### 1.3 Arquitectura del sistema.

En la arquitectura del sistema vemos como se conectan todas esas antes descritas, pasamos de ver un proyecto formado por varios componentes a un componente global, en nuestro caso TOP\_ASCENSOR1.



Al ser tantos componentes e interconexiones hemos hecho una a mano para que se vea mejor.



## 1.4 Verificación funcional.

Una vez hemos realizado nuestro primer intento de diseño tenemos que comprobar que el diseño realmente cumple los requisitos establecidos y se amolda a lo deseado.

Esto lo haremos a través de simulaciones de cada componente que añadiremos a la rama test en el control de versiones de github.

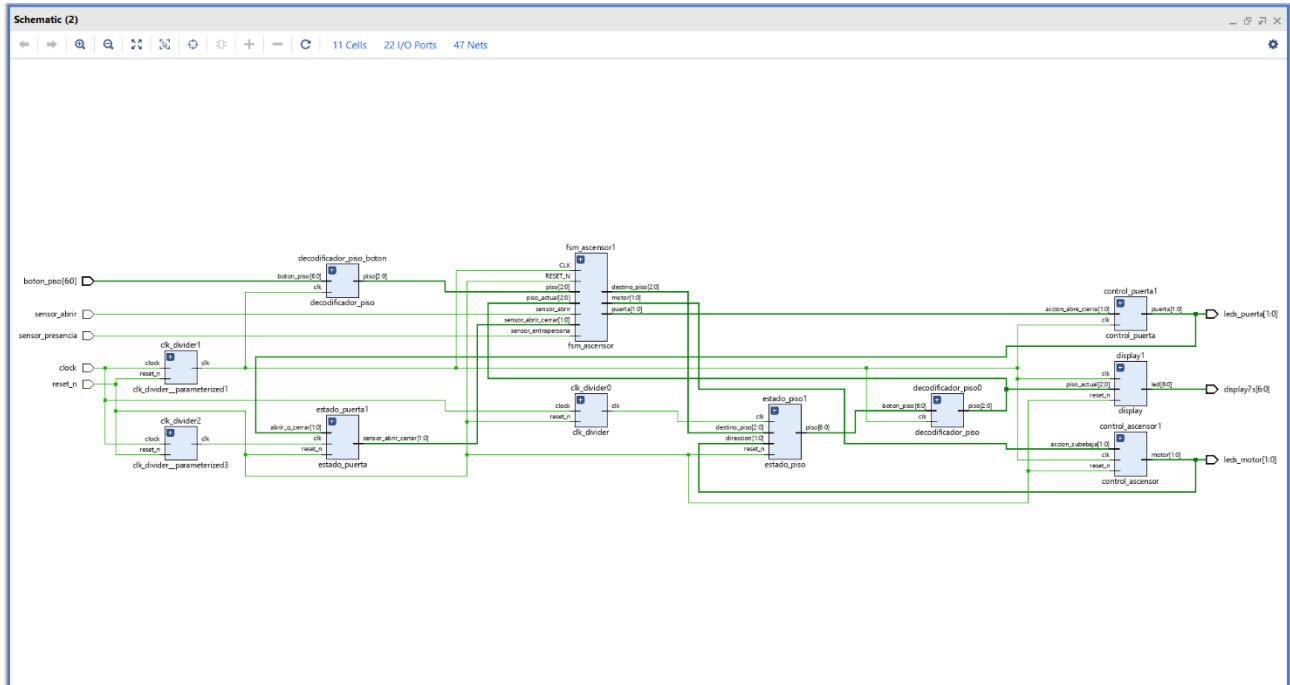
## 1.5 Cierre de pre-síntesis

Una vez hayamos simulado todos los componentes y hayamos visto que no hay errores ni sintácticos ni de síntesis podremos seguir desarrollado el diseño.



## 1.6 Síntesis

Este proceso es fundamental ya que sirve para refinar y depurar el proceso para que quede lo más limpio y correcto posible.



## 1.7 Proceso de verificación post-síntesis

En nuestro caso es inmediato ya que como no hemos añadido nuevas restricciones después de la síntesis la verificación es efectiva de inmediato.

## 1.8 Cierre de diseño

Por último, podemos declarar el cierre del diseño ya que:

1. El diseño cumple todas las restricciones (en nuestro caso se limite a las únicas que tenemos que son las iniciales).
2. Se cumplen las especificaciones temporales.

## 2 Problemas ocasionados y soluciones adaptadas.

El primer problema con el que nos encontramos fue como podríamos hacer que cada parte del diseño funcionase a frecuencias diferentes si cogíamos la misma señal del reloj de la placa.

Tras búsqueda en internet y en los apuntes vimos que podríamos hacer una división de la frecuencia del reloj original de la placa adecuada a la necesaria para cada dispositivo.

Como tendríamos que realizar este proceso varias veces vimos que la mejor manera era realizando un componente cuya única función fuese dividir la frecuencia del reloj de la placa, para que sirviese con distintas frecuencias a distintos módulos o dispositivos.

El segundo problema que tuvimos que abordar fue la sincronización de control puerta y estado puerta al igual que control ascensor y estado piso. Aunque inicialmente parecía que sus funcionalidades estaban muy definidas, al ir implementando todos los módulos, se fue disipando esa claridad de ideas y pusimos demasiado peso en ambos módulos y se pisaban la funcionalidad entre ellos.

Esto lo pudimos solucionar dejando que los módulos de control se limitasen solamente a pasar información de la FSM a los módulos de estado que eran los que de verdad manejaban las salidas de motor y puerta.

## 3 Conclusión.

Al principio intentamos abordar este trabajo con un método o forma down-top en el cual intentábamos hacer cada módulo desde cero para luego realizar la parte estructural juntando todos los módulos y tener la top completa. Gracias a las tutorías vimos que este método era erróneo ya que perdíamos la claridad de la idea intentando realizar todo a la vez desde un principio, además de que esto ocasionaba muchos errores.

Finalmente optamos por el método top-down en el cual primero meditamos bien que íbamos a realizar y que requisitos necesitábamos, luego casi todo el trabajo fue la separación de cada tarea en sus bloques y ver como podíamos minimizar al máximo la influencia de la fsm (ya que en un principio todo lo hacía este módulo). De esta forma iniciamos haciendo el diagrama de bloques y viendo la funcionalidad de cada uno, posteriormente hicimos unos testbench muy sencillos para cada funcionalidad pensada para ellos y así finalmente realizamos el código de cada módulo de manera mucho más sencilla.

En conclusión, merece mucho la pena en destinar mucho tiempo solo a pensar y tener claros los requisitos y las funcionalidades porque luego esto facilitará infinitamente el trabajo de código.

## 4 Enlaces a github y al video demostrativo.

[Video demostrativo trabajo SED VHDL-ascensor](#)

<https://www.youtube.com/watch?v=IwjHK0SvR-w>



Enlace de github control de versiones.

<https://github.com/mariaortegamonge/trabajo-sed-vhdl>