

TRABAJO

S.E.D

ADRIÁN ARROYO JIMÉNEZ	51089
DAVID GRANDE GIL	51249
JOSE LUIS GRANDE MORÓN	51250

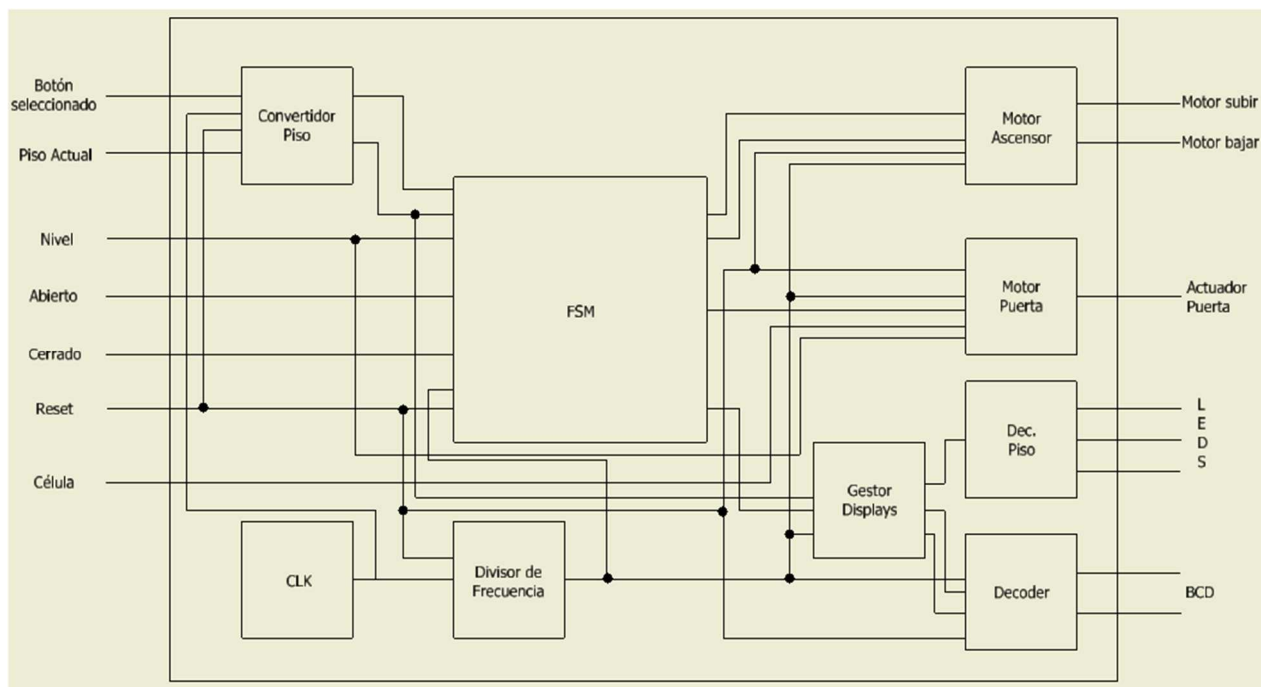
ÍNDICE

1. DESCRIPCIÓN DEL TRABAJO
2. DIAGRAMA DE BLOQUES
3. ESTRATEGIA SEGUIDA Y SOLUCIONES ADOPTADAS

1. DESCRIPCIÓN DEL TRABAJO

El proyecto consistía en diseñar el controlador de un ascensor en una vivienda de 3 pisos. El funcionamiento del ascensor consiste en ir al piso indicado por los botones, abrir las puertas y esperar a otra llamada (se hace caso omiso a estas llamadas si el ascensor está en movimiento). En caso de que haya un obstáculo en las puertas no se cerrarán. Toda la información del piso en el que se encuentra el ascensor y su funcionamiento será mostrada por los Displays y LEDs.

2. DIAGRAMA DE BLOQUES



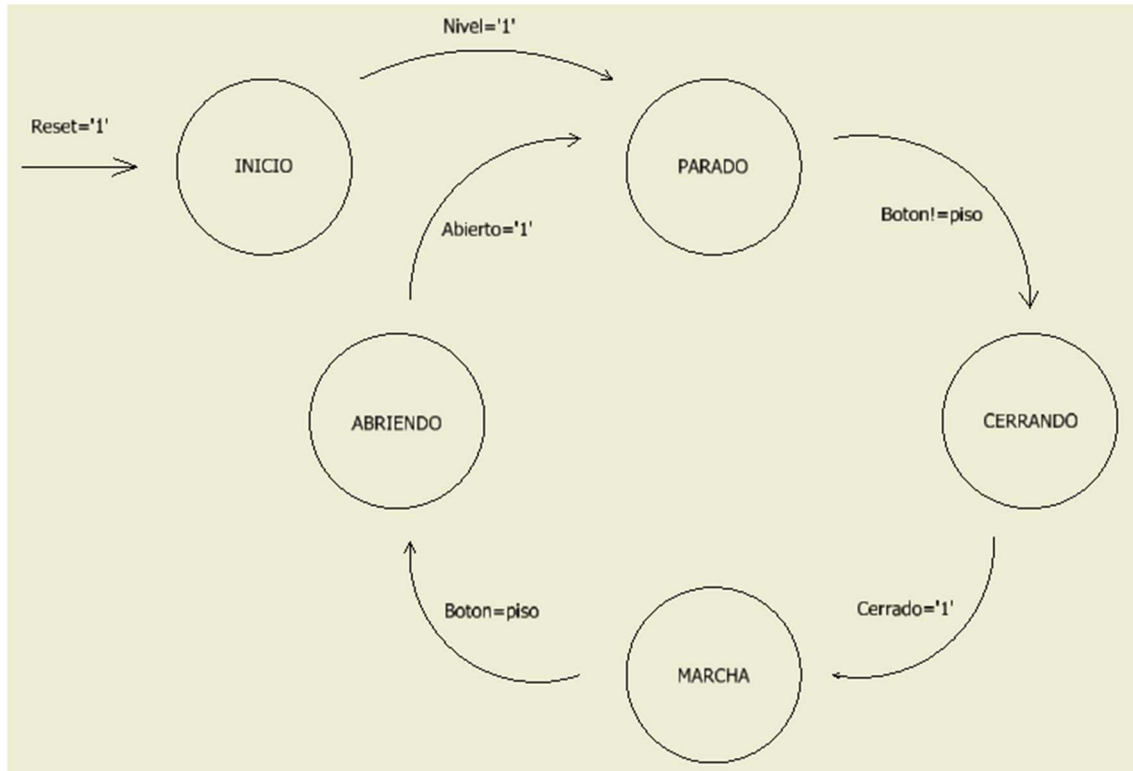
BLOQUE "CONVERTIDOR PISO":

Este bloque es el encargado de recibir la información del piso en el que se encuentra el ascensor, la planta a la que se desea ir, y transformarlo en información útil y manejable para el bloque de estados. Recibe como entradas dos vectores de tres bits correspondientes a el piso en el que se encuentra el ascensor y al piso al que se desea ir, una entrada del reloj, y otra entrada correspondiente al reset. Da como salidas la codificación del piso actual convertido y el piso seleccionado convertido.

Dentro de este bloque se han incluido métodos antirebote para cada uno de los 3 pulsadores que tenemos con el fin de evitar falsas pulsaciones.

BLOQUE "FSM":

Este bloque es el encargado de gestionar nuestra máquina de estados. Recibe como entradas la información de un sensor que nos permite saber si la puerta está abierta o cerrada, la posible pulsación del reset, la información de nivel del ascensor y en que piso se encuentra el ascensor y hacia dónde se desea ir. Tiene como salidas las órdenes a los motores y al gestor de displays. El bloque está compuesto por cinco estados:



-Inicio: Estado cuya función sirve para cuando se enciende el circuito o se hace reset. Permite llevar al ascensor a un piso cualquiera, de lo contrario se podría parar entre dos pisos.

-Parado: Estado que se encuentra a la espera de que se le pulse un botón para ir al piso indicado. Es el estado en el que normalmente se encontrará el ascensor.

-Cerrando: Estado encargado de gestionar el cierre de la puerta del ascensor. Debe comprobar que no hay ningún obstáculo y que se permite cerrar la puerta.

-Abriendo: Estado encargado de gestionar la apertura de las puertas del ascensor.

-Marcha: Estado en el que se encuentra el ascensor para ir de un piso a otro.

Además este bloque se encarga de gestionar la memoria de la pulsación de los botones mientras que el ascensor está haciendo una tarea, es decir, mientras que no se encuentre en el estado de parado. Aunque por orden de enunciado el ascensor debe hacer caso omiso a la pulsación de botones mientras se encuentre ejecutando una tarea, es necesario capturar la pulsación del botón, así como el piso donde se encuentra. Sólo se captura la pulsación correcta, por ello, hay que hacer esta comprobación, puesto que podría darse el caso de pulsar dos botones a la vez.

BLOQUE “DIVISOR DE FRECUENCIAS”:

Bloque encargado de dividir la frecuencia de entrada del reloj en una relación.

BLOQUE “GESTOR”:

Este bloque es el encargado de gestionar los distintos decodificadores de los displays.

BLOQUE “DECODIFICADOR PISO”:

Este bloque es el encargado de recibir la orden del número en el que se encuentra el ascensor, y transformarlo para que se encienda el led asignado a ese piso.

BLOQUE “DECODER”:

Este bloque es el encargado de recibir la orden de que el ascensor está en movimiento, y transformarlo en una interfaz formada por flechas visto desde el display.

BLOQUE “MOTOR ASCENSOR”:

Este bloque es el encargado de mover el motor del ascensor. Recibe una entrada de un bit indicando si el motor debe subir o bajar, y tiene como salidas la orden de que haga girar el motor del ascensor en una u otra dirección.

BLOQUE “MOTOR PUERTA”:

Este bloque es el encargado de mover el motor de la puerta del ascensor. Recibe dos entradas, una de un bit indicando si las puertas deben abrirse o cerrarse y la otra indicando si hay algún obstáculo en la puerta. En caso de que haya un obstáculo por motivos de seguridad la puerta no puede cerrarse. Como salida se tiene la orden al actuador de la puerta.

3. ESTRATEGIA SEGUIDA Y SOLUCIONES ADOPTADAS

La estrategia sobre la que hemos basado el desarrollo del proyecto ha sido el encapsulamiento. Hemos ido adoptando medidas para obtener una creación y un diseño de bloques simple y que las funciones fueran llevando la solución de lo más general a lo más particular. De ahí que en la creación y el esquema de los bloques haya varios encargados de la gestión y el control de otros bloques.

Para la creación del código del proyecto se ha usado la plataforma de desarrollo colaborativo GitHub, con el fin de que nos fuera más fácil y eficiente el desarrollo del código.

Un punto importante en el desarrollo del proyecto fue la elección de los estados de nuestra máquina de estados. En un principio pensamos en que cada uno de los cuatro pisos fuera un estado, pero pronto nos dimos cuenta de que eligiendo como estados su estado en cuanto a movimiento se refiere (parado, marcha, cerrando...) era mucho más extrapolable e intuitivo, ya que nos permitiría llevar este módulo a un ascensor con irrelevancia del número de pisos que tuviera.

En el desarrollo del proyecto, otro aspecto valorado y que finalmente se ha incluido es la creación de un bloque divisor de frecuencia, ya que para determinadas acciones nos interesaba una frecuencia menor para no gastar tantos recursos. Sin embargo para otras acciones nos compensaba usar la frecuencia de la FPGA, como por ejemplo en nuestro caso los métodos antirebote. Desde un primer momento creímos que era indispensable incluir un método antirebote para los pulsadores, por lo que lo implementamos con el fin de evitar pulsaciones falsas.