

Proyecto 3: Diseño de las Máquinas de Estados
Diseño de un Controlador de Motor Paso a Paso
Trabajo realizado por:
Arnau Mora Gras & Carlos Villena Jiménez
amorgra@teleco.upv.es & cviljim@teleco.upv.es

Índice:

Introducción.

Subtarea 1 → Controlador de Motor Paso a Paso.

Subtarea 2 → Verificación del controlador en hardware.

Subtarea 3 → Rediseño del proyecto 1 (Juego De Luces).

Conclusión.

I) Introducción →

En este proyecto diseñaremos un circuito controlador para un motor paso a paso, basado en una máquina de estados finitos, y comprobaremos su funcionamiento mediante un emulador de motor.

¿Qué es un motor a paso? →

Es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de avanzar una serie de grados (paso) dependiendo de sus entradas de control.

Para un motor de este tipo hay tres posibles secuencias de accionamiento:

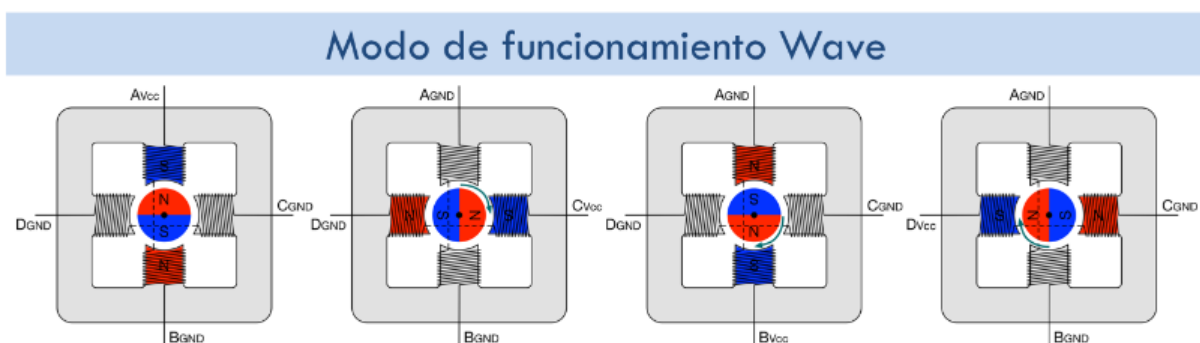


Fig1. Modo funcionamiento de motor wave o onda completa.

Modo de funcionamiento Normal

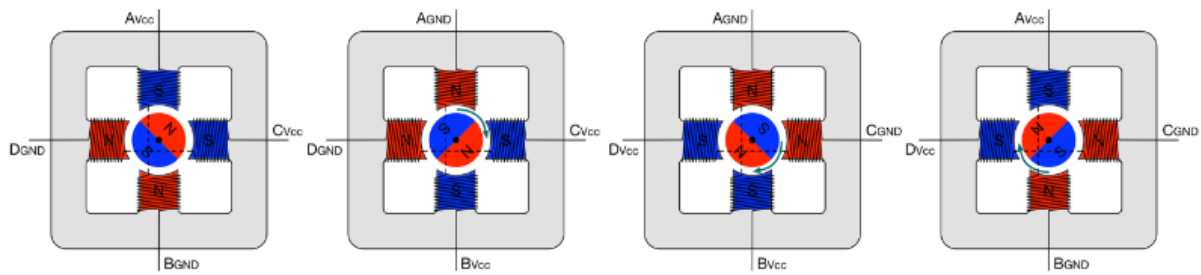


Fig2. Modo funcionamiento de motor normal o paso completo.

Modo de funcionamiento Medio Paso

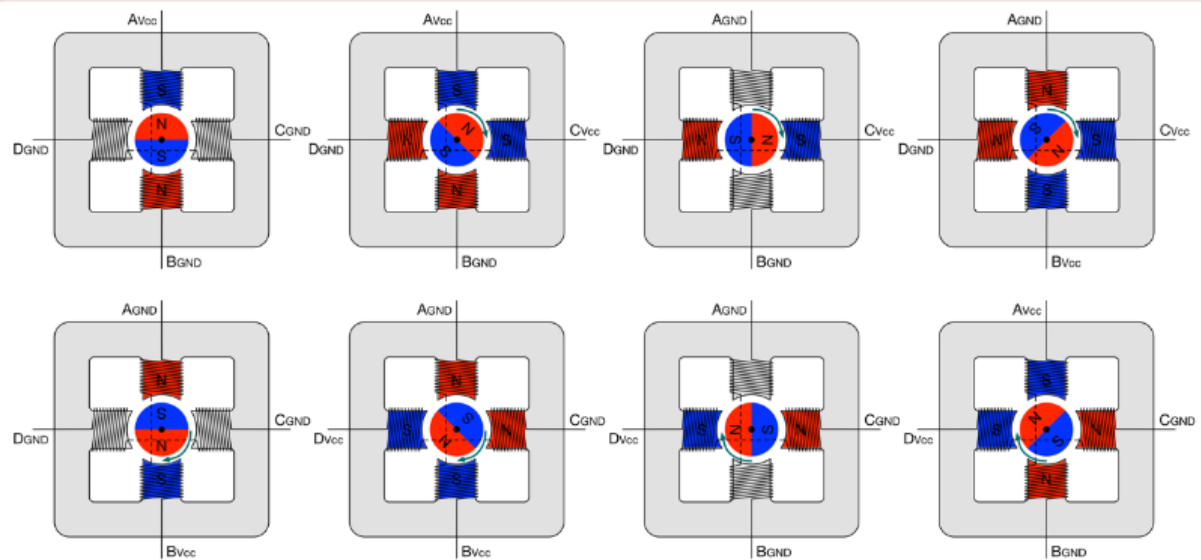


Fig3. Modo funcionamiento de motor a medio paso.

Subtarea1. Controlador de motor Paso a Paso.

En este apartado se pretende realizar el diseño de un módulo para la generación del movimiento del motor paso a paso teniendo en cuenta las posibilidades de movimiento según las entradas que se ven en la figura 4.

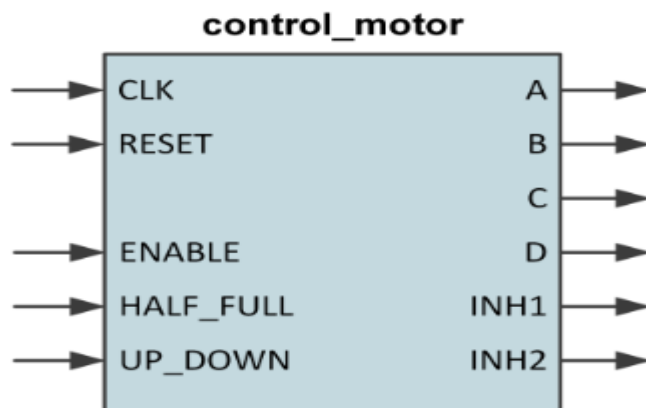


Fig4. Bloque del módulo del contro_motor.v

Donde podemos observar la influencia de 5 entradas:

- **CLK**, línea de reloj, activa por flanco de subida.
- **RESET**, asíncrono y activo a nivel bajo, posiciona el controlador en el estado inicial, también llamado "HOME".
- **ENABLE**, activa a nivel alto, habilita el funcionamiento del controlador; es decir, mientras esté activa el controlador debe hacer girar el motor (ir cambiando de estado).
- **UP_DOWN**, esta línea sirve para indicar el sentido de giro del motor. Cuando está a nivel alto el motor gira en el sentido de las agujas del reloj, mientras que a nivel bajo lo hace en sentido contrario.
- **HALF_FULL**, esta línea permite configurar el modo de funcionamiento. Cuando está a nivel alto el motor está en modo HALF, mientras que a nivel bajo el motor se configura en modo NORMAL o WAVE.

y consta también de 6 entradas, las cuales determinaría el modo de funcionamiento detallado en las tres primeras figuras.

Básicamente y de forma escrita, sería tal que así:

- el modo "HALF" o de medio paso en el que la máquina pasa secuencialmente por todos los estados,
- el modo "NORMAL", en el que el controlador únicamente pasa por los estados impares,
- y por último el modo "WAVE" en el que la evolución se realiza a través de los estados pares.

En cuanto al código del diseño del módulo, se realizará mediante una FSM con tres bloques procedurales y seguirá el siguiente diagrama de bloques.

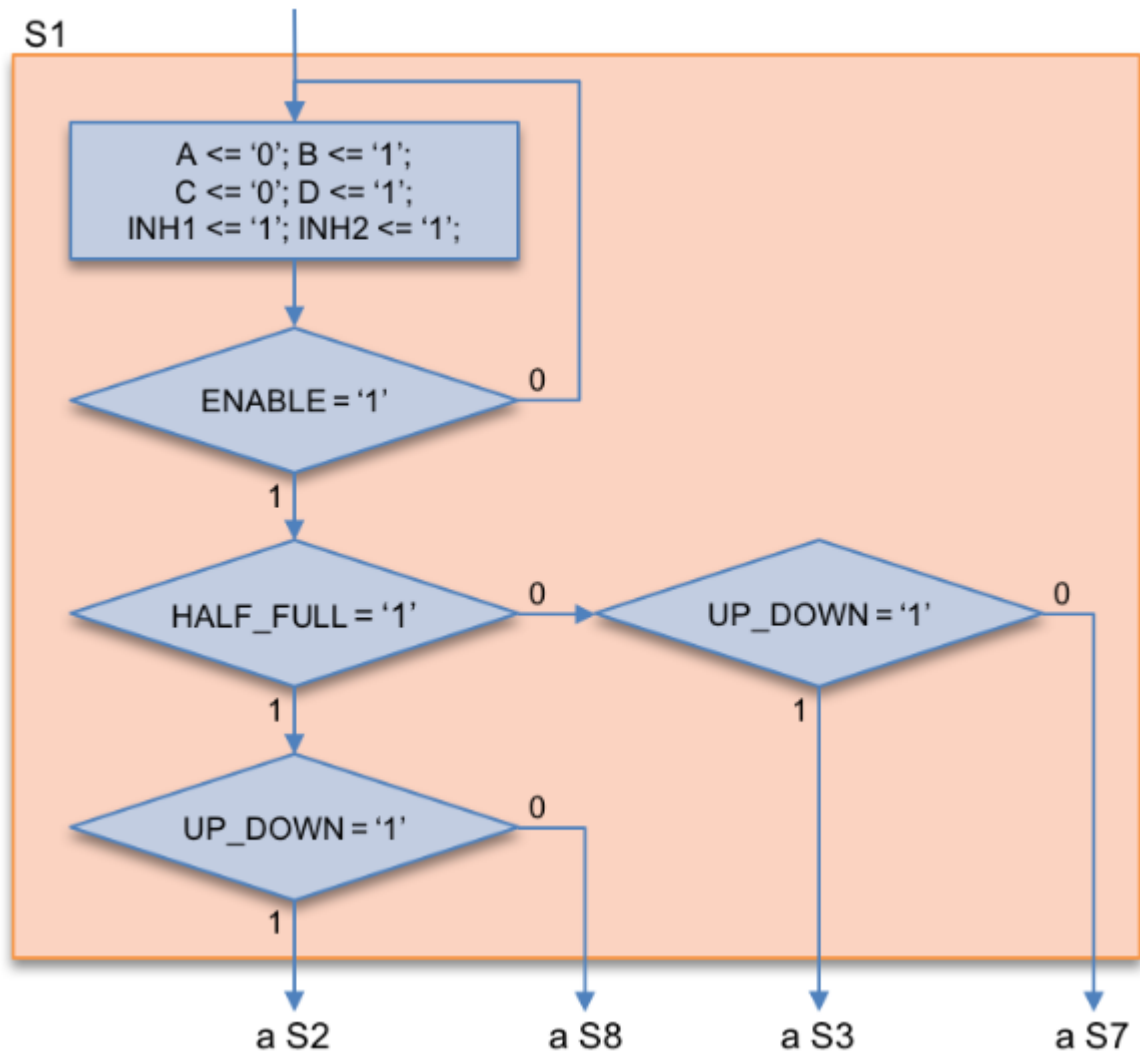


Fig5. Diagrama de bloques para el módulo control_motor.

En nuestro caso, el código ha seguido la siguiente estructura:

Primer always para la actualización de los estados.

Segundo always para la asignación de estado actual dependiente de las señales de entrada, como las de giro derecha/izquierda, modo de funcionamiento...

Último bloque always para la asignación de las salidas dependiendo del estado en el que se encuentre.

A esta forma de implementación FSM se le conoce como máquina de MOORE.

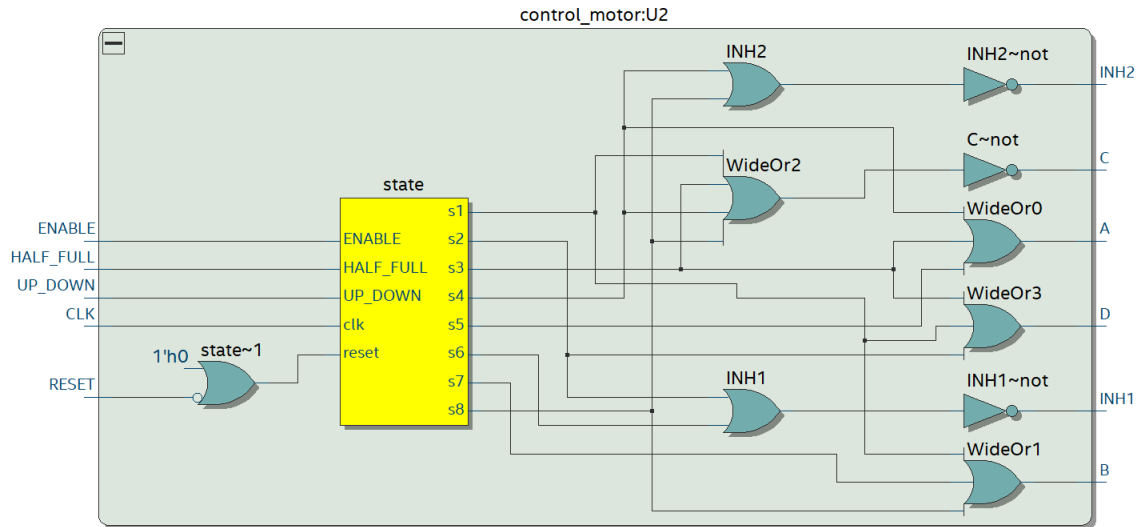


Fig5.2. RTL viewer sobre el módulo control_motor.v

Subtarea2. Verificación del controlador en hardware.

Para este apartado seguiremos de forma sencilla, descargando un proyecto ya creado por los profesores para realizar las funciones y visualizaciones del diseño de forma rápida sobre los leds y una pantalla LED.

El esquema de implementación a seguir es el siguiente:

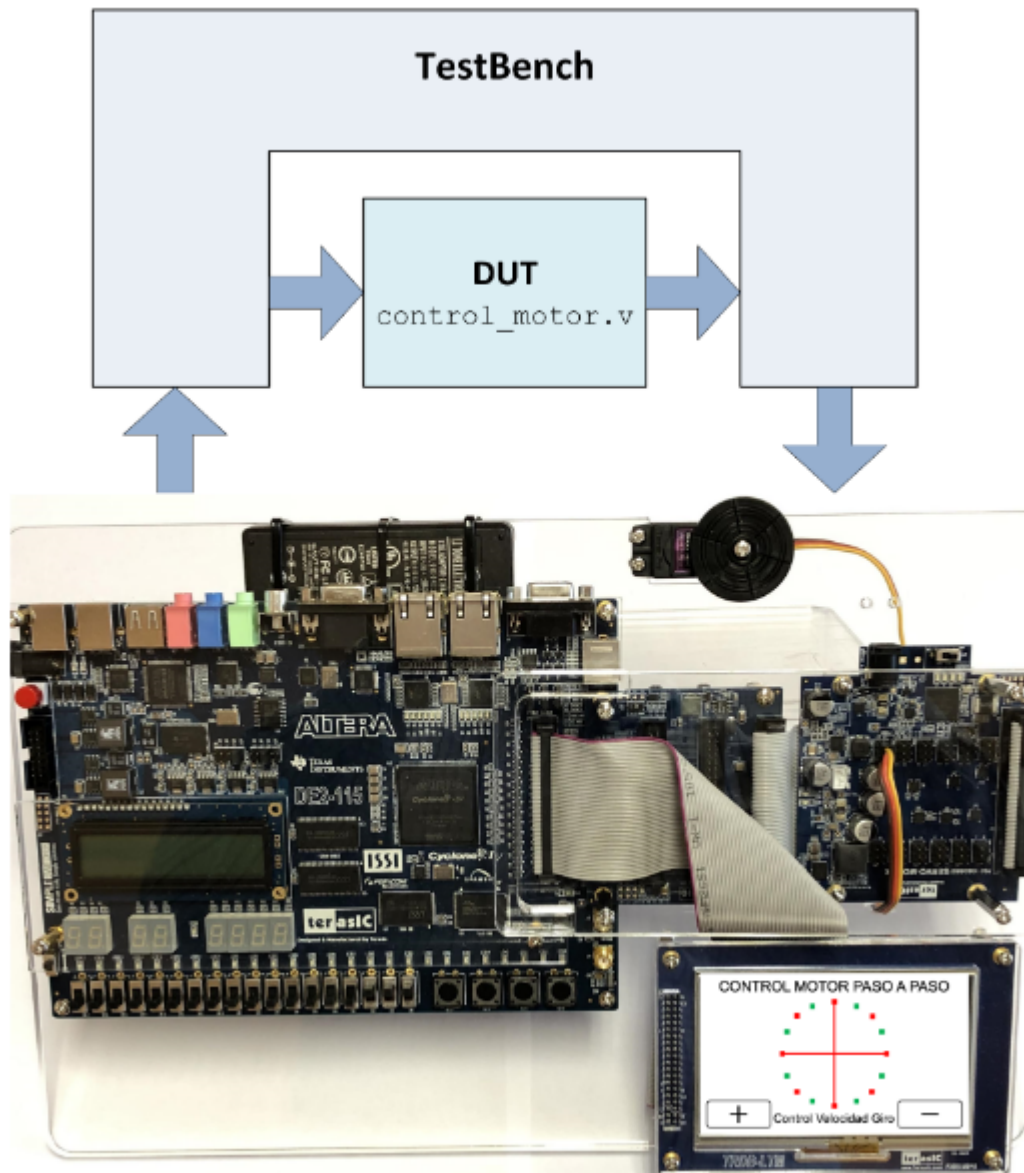


Fig6. Diagrama de Bloques para implementación de la verificación.

Subtarea 3. Rediseño del Proyecto 1 (Juego de luces).

Lo que se pretende en este apartado es la realización de juego de luces que se realizó en la primera práctica del curso (Juego de luces), pero de forma en la que nos ayudemos de esto. Aparte se nos pide que diseñemos también un control de velocidad .

En la siguiente imagen se puede observar el diagrama de bloques que se debe de implementar en módulos.

Juego de luces con velocidad variable

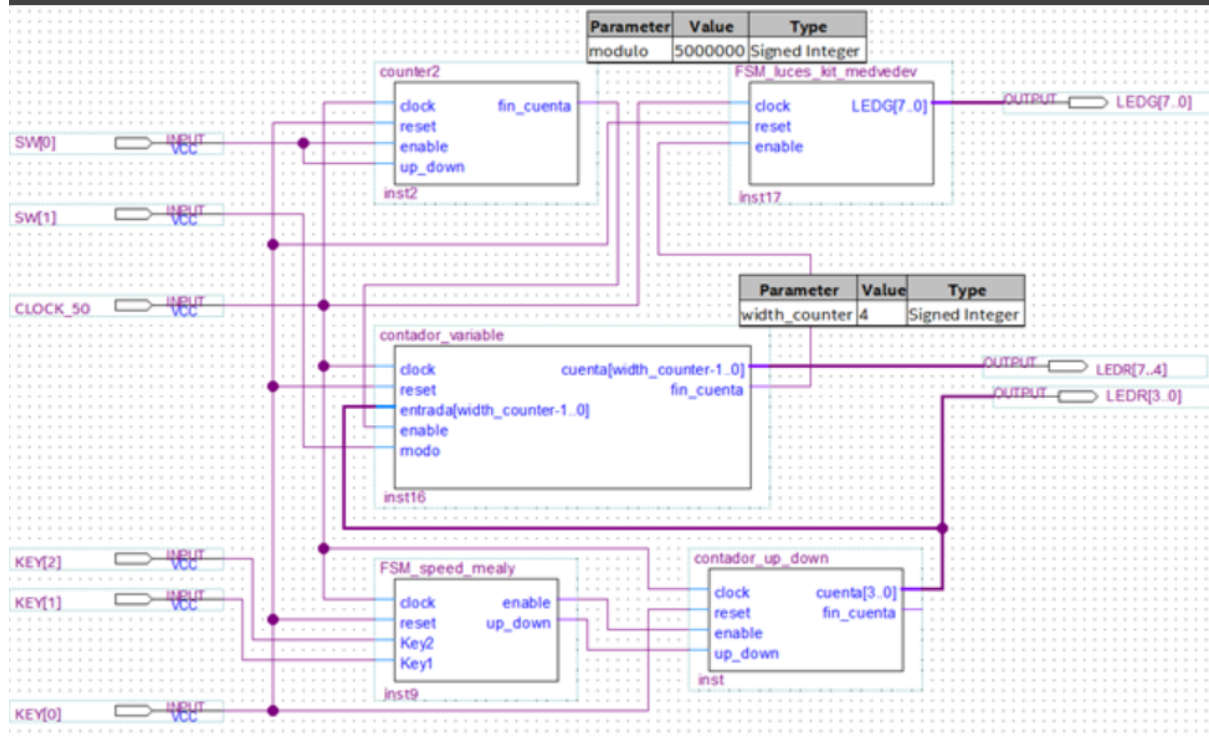


Fig7. Diagrama de Bloques para la parte final.

La explicación de cada uno de los módulos a implementar se detalla especialmente en el encabezado de cada código sobre verilog en formato comentarios '%'.

Pero si que detallaremos unas imágenes sobre los testbench realizados a algunos de los módulos para que se vea su correcto funcionamiento.

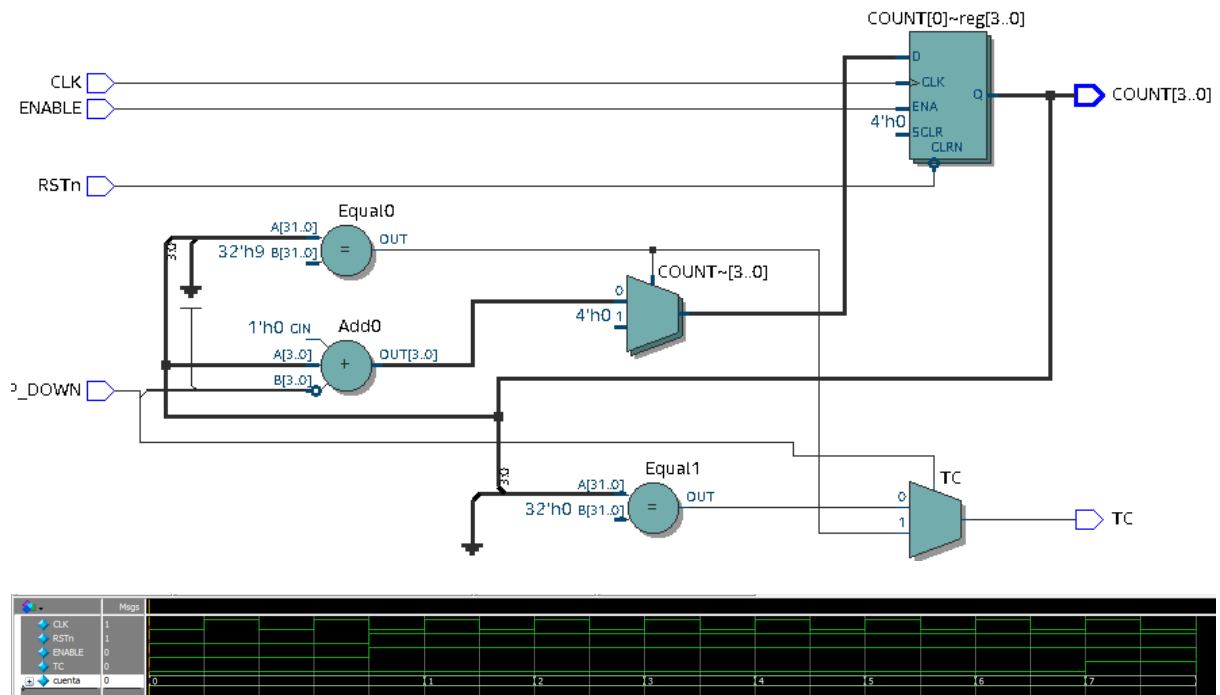


Fig8. RTL viewer y waveform del testbench del módulo contador.v

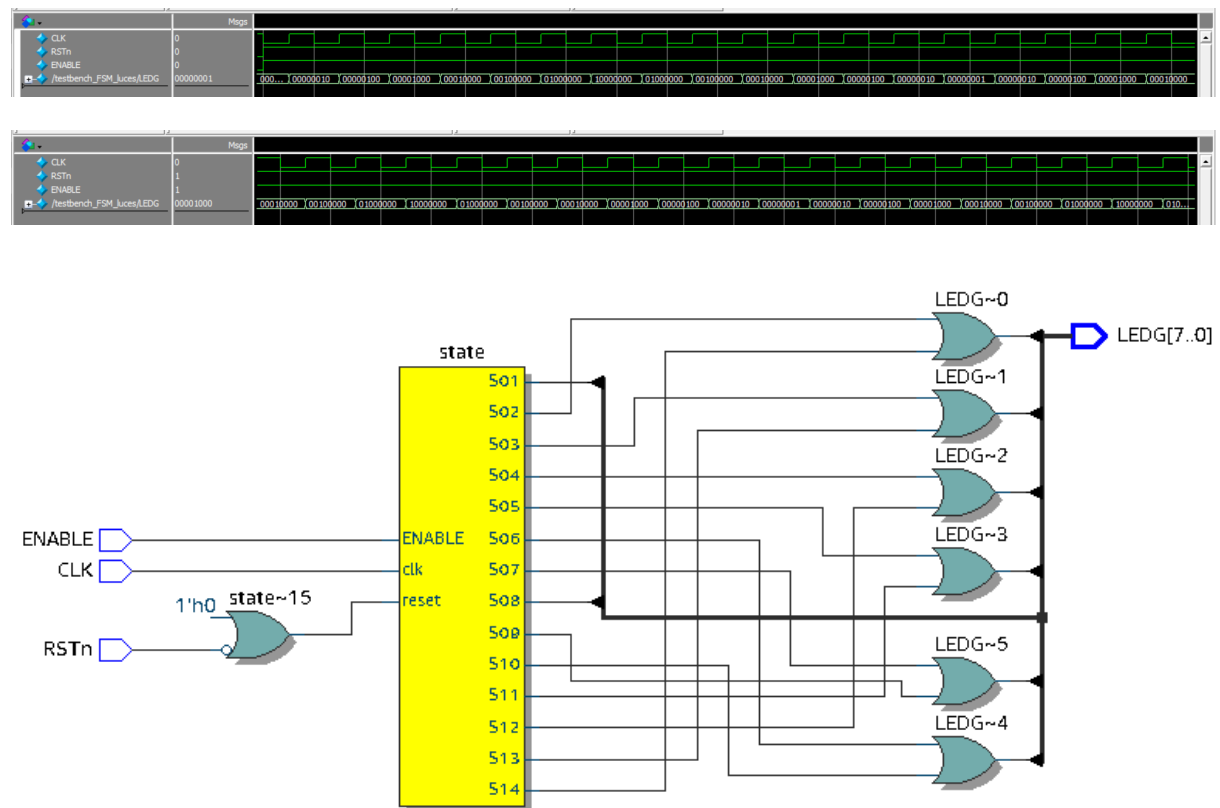


Fig9. RTL viewer y un par de waveform testbench sobre el móduloFSM_luces.v

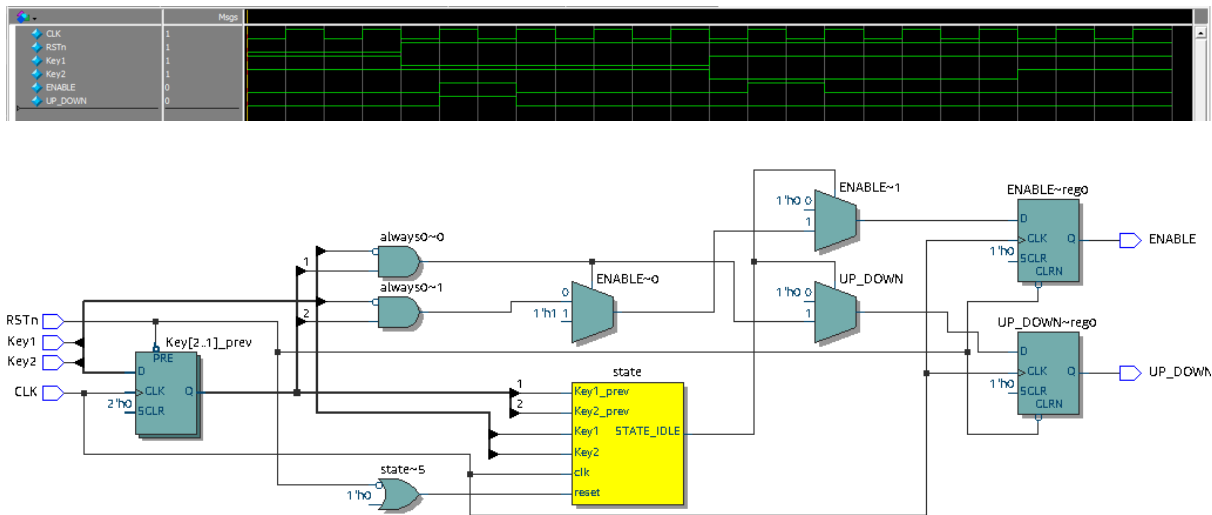


Fig10. RTL viewer y wave form testbench sobre módulo FSM_speed.v

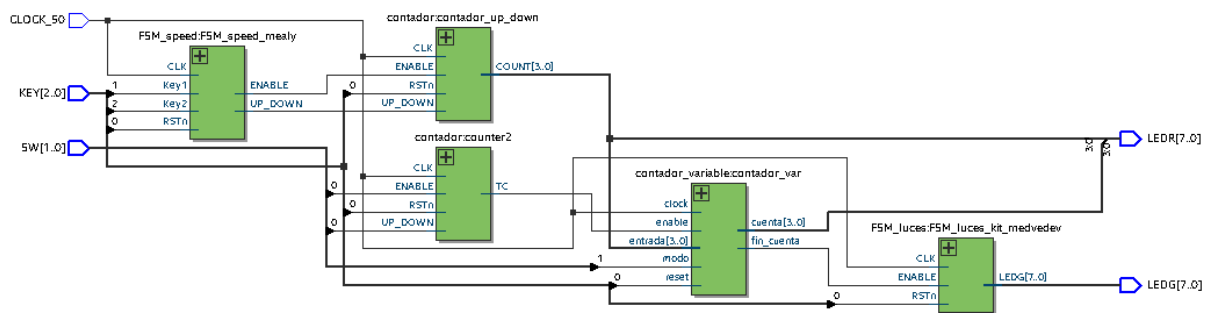


Fig11. RTL viewer del módulo juego_luces_vel_var.v

Luego la asignaciones de pines que se han hecho han ido de la mano de la siguiente imagen:

Nombre Puerto	Pin FPGA	Nombre Puerto	Pin FPGA	Nombre Puerto	Pin FPGA
CLOCK_50	Y2	LEDG[0]	E21	LEDR[0]	G19
KEY[0]	M23	LEDG[1]	E22	LEDR[1]	F19
KEY[1]	M21	LEDG[2]	E25	LEDR[2]	E19
KEY[2]	N21	LEDG[3]	E24	LEDR[3]	F21
KEY[3]	R24	LEDG[4]	H21	LEDR[4]	F18
SW[0]	AB28	LEDG[5]	G20	LEDR[5]	E18
SW[1]	AC28	LEDG[6]	G22	LEDR[6]	J19
SW[2]	AC27	LEDG[7]	G21	LEDR[7]	H19

Fig12. Asignación de pines sobre las variables.

Conclusión.

En este tercer proyecto hemos llegado a entender correctamente lo que es un motor, en sí un motor paso a paso, y las formas de aplicarlo que se puede llegar a tener

A su vez, se ha vuelto a desarrollar un mismo trabajo ya hecho del desplazamiento de unos leds de derecha a izquierda o viceversa, pero mediante la aplicación del motor a pasos, e incluyendo la posibilidad de cambios de velocidad.