



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

**PROYECTOS DE IMPLEMENTACIÓN DE FUNCIONES LÓGICAS
FUNDAMENTALES**

LABORATORIO 3

DISEÑO VLSI

Autores: David Carneros Prado
María Espinosa Astilleros

Fecha: 4 de marzo de 2020

Índice

1. Proyecto and_2	2
1.1. Creación proyecto	2
1.2. Diseño del circuito lógico	2
1.3. Configuración de pines	3
1.4. Simulación del circuito lógico	3
1.5. Errores producidos en la realización del proyecto	4
2. Proyecto gates2	5
2.1. Código fuente	5
2.2. Configuración de los pines	6
2.3. Tabla de verdad	6
2.4. Funcionamiento del proyecto	7
2.5. Simulación del proyecto	8
2.6. Errores producidos en la realización del proyecto	8

1. Proyecto and_2

En este proyecto realizaremos una puerta AND con dos entradas conectadas a los pulsadores de la tarjeta DE0-Nano y una salida conectada a uno de los leds de dicha tarjeta.

1.1. Creación proyecto

Para crear el proyecto de diseño mediante captura de esquemáticos debemos seleccionar la opción *Block Diagram/Schematic File*.

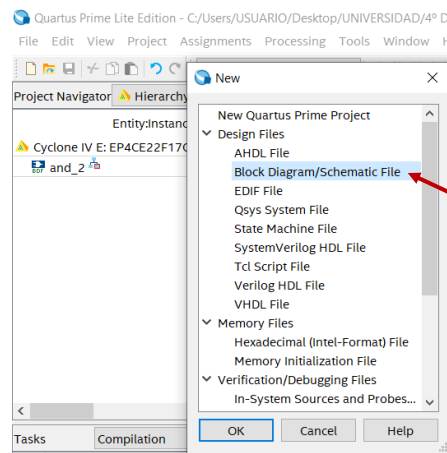


Figura 1: Creación diseño del circuito lógico

1.2. Diseño del circuito lógico

Una vez creado el proyecto diseñamos el circuito lógico que hemos explicado anteriormente al inicio de esta sección. Al terminar de diseñar el circuito tendremos que compilar el diseño seleccionando la opción *Analysis & Synthesis*.

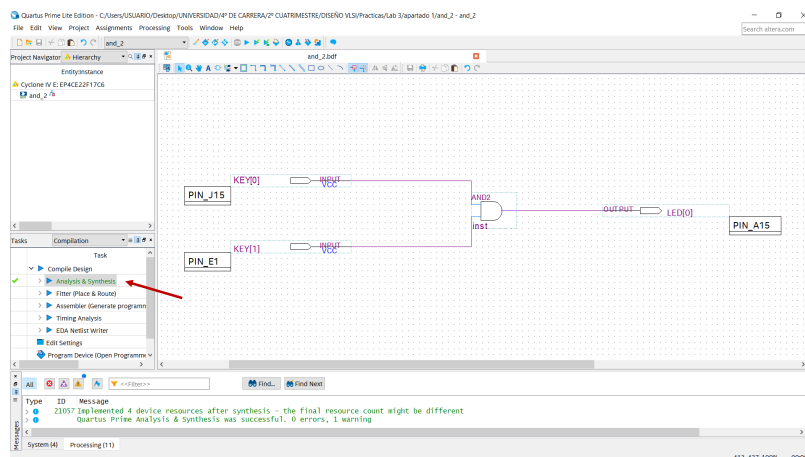


Figura 2: Diseño y compilación del circuito lógico

1.3. Configuración de pines

Por último, configuramos los pines que se nos indica en el enunciado. En este caso seleccionamos el pin J15 y el pin E1 para las entradas de los pulsadores KEY[0] y KEY[1] respectivamente, y el pin A15 para la salida del LED[0]. Para ello seleccionamos el icono del *Pin Planner* y escribimos en cada uno de los nodos la localización correspondiente.

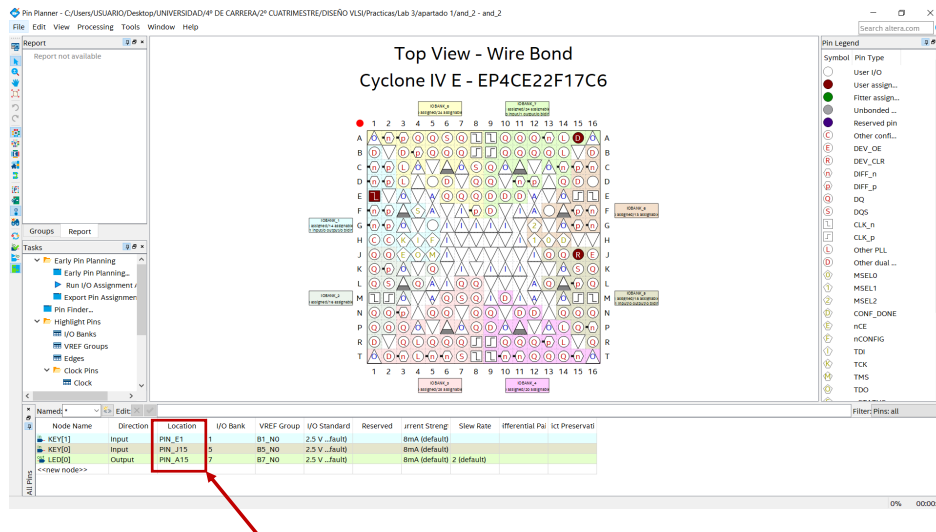


Figura 3: Configuración de los pines

1.4. Simulación del circuito lógico

En este apartado realizaremos una simulación de nuestro circuito. En primer lugar crearemos el archivo de simulación seleccionando la opción *University Program VWF*

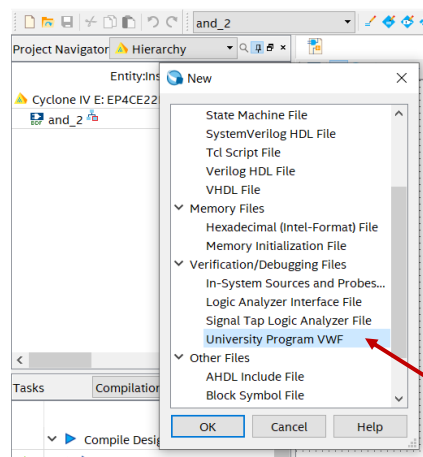


Figura 4: Creación del archivo de simulación

Listamos los nodos, que son KEY[0], KEY[1] y LED[0], y los configuramos para que KEY[0] tenga un periodo cada 2 μ s y KEY[1] tenga un periodo cada 1 μ s

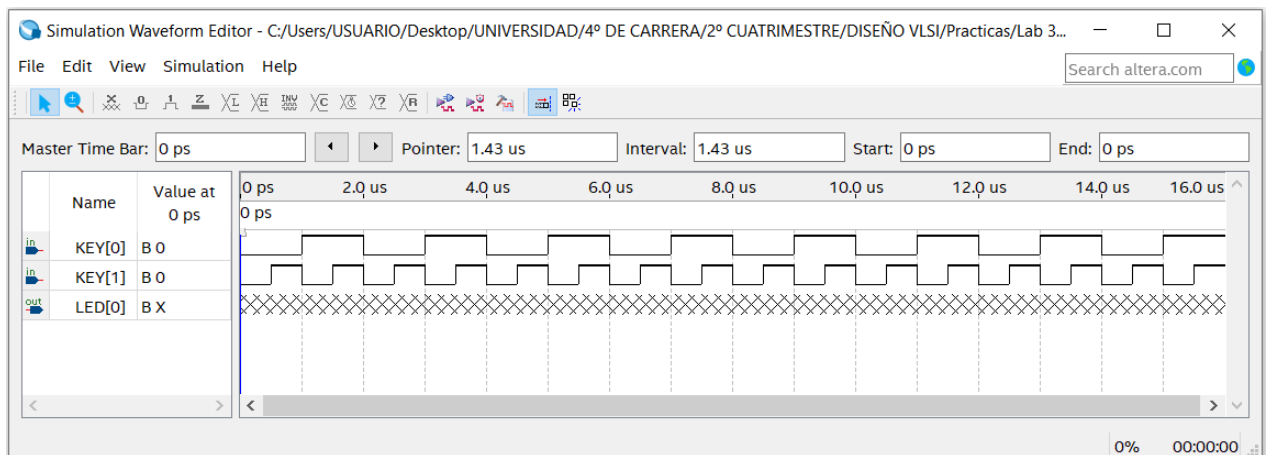


Figura 5: Configuración de los nodos

Finalmente ejecutamos la simulación seleccionando la opción *Run Functional Simulation* de la pestaña *Simulation*.

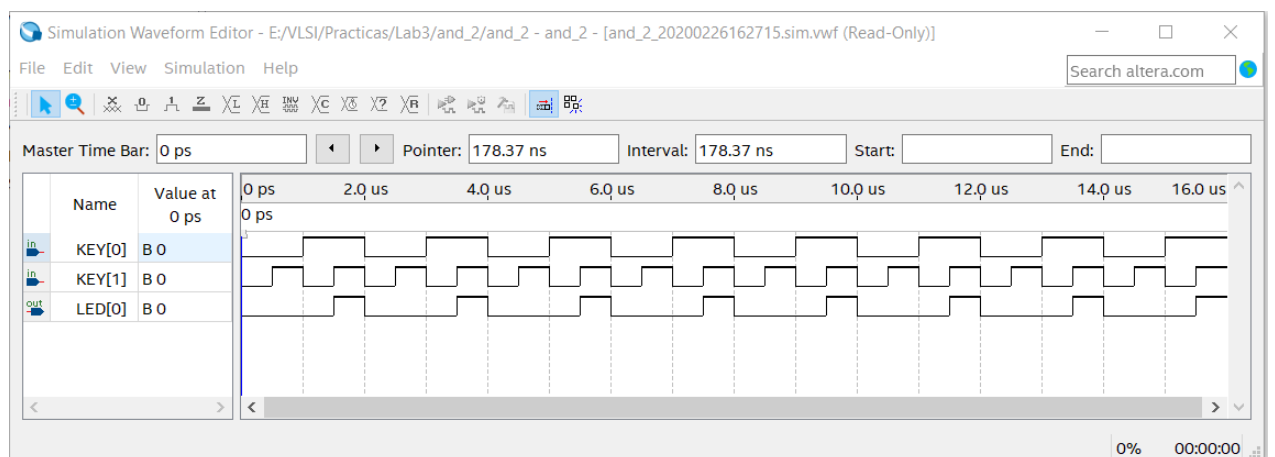


Figura 6: Simulación del proyecto

Como se puede observar en la Figura 6, el LED[0] estará encendido sólo cuando los pulsadores KEY[0] y KEY[1] estén activos simultáneamente.

1.5. Errores producidos en la realización del proyecto

En la realización del proyecto nos han surgido varios errores, los cuales comentaremos e indicaremos cómo los hemos solucionado:

- Al crear el proyecto con el nombre *and2* se producía un error porque la puerta AND que usabamos se llamaba de la misma manera. Para solucionarlo simplemente cambiamos el nombre del proyecto por *and_2*
- Al ejecutar la simulación se producía un error por el nombre de la ruta donde teníamos el archivo guardado, ya sea por los espacios o por los caracteres especiales. Para ello cambiamos la ruta donde guardamos los archivos.

2. Proyecto gates2

En este proyecto realizaremos la implementación de tres puertas lógicas básicas (AND, OR, XOR) de 2 entradas conectadas respectivamente a los switches DIP de la tarjeta DE0-Nano. Las salidas de las puertas se conectarán a tres LED de salida (LED[0], LED[1], LED[2]) y adicionalmente las entradas se conectarán a dos de los LED de entrada (LED[6], LED[7]).

Este proyecto de diseño estará implementado mediante VHDL.

2.1. Código fuente

Listing 1: Código fuente del proyecto gates2

```
1  -- Laboratorio 3: Proyectos de implementación de funciones lógicas fundamentales
2  -- Autores: David Carneros Prado y Maria Espinosa Astilleros
3  -- Desarrollo: Implementación de tres puertas lógicas básicas (AND, OR, XOR) de 2 entradas conectadas
4  -- respectivamente a los switches DIP de la tarjeta DE0-Nano.
5  -- Las salidas de las puertas se conectarán a tres LED de salida (LED[0], LED[1], LED[2]) y
6  -- adicionalmente las entradas se conectarán a dos de los LED de entrada (LED[6], LED[7])
7
8
9  LIBRARY ieee;
10 USE ieee.std_logic_1164.ALL;
11
12 ENTITY gates2 IS
13     PORT (
14         SW : IN STD_LOGIC_VECTOR(0 TO 1);    -- dip switches
15         LED : OUT STD_LOGIC_VECTOR(0 TO 7));   -- green leds (LED[0], LED[1], LED[2],
16                                                -- LED[6] y LED[7])
17
18 END gates2;
19
20 ARCHITECTURE LogicFunc OF gates2 IS
21 BEGIN
22     LED(0) <= SW(0) AND SW(1);    -- puerta AND conectada al LED[0]
23     LED(1) <= SW(0) OR SW(1);    -- puerta OR conectada al LED[1]
24     LED(2) <= SW(0) XOR SW(1);    -- puerta XOR conectada al LED[2]
25     LED(6) <= SW(0);              -- SW[0] conectado al LED[6]
26     LED(7) <= SW(1);              -- SW[1] conectado al LED[7]
27 END LogicFunc;
```

2.2. Configuración de los pines

Configuramos los pines como nos indica el enunciado.
Para los switches SW[0] y SW[1] seleccionamos los pines M1 y T8, respectivamente y para los leds LED[0], LED[1], LED[2], LED[6] y LED[7] seleccionamos los pines A15, A13, B13, B1 y L3, respectivamente.

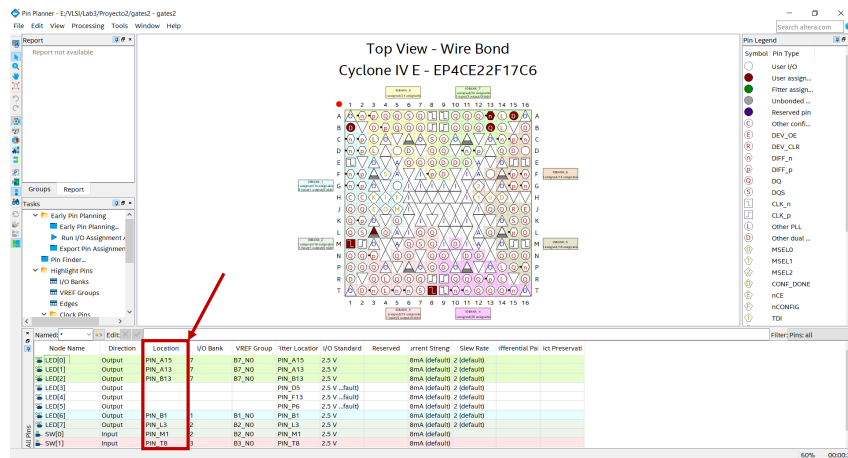


Figura 7: Configuración de los pines

2.3. Tabla de verdad

Partiendo de los datos del enunciado hemos creado la tabla de verdad del proyecto, quedando de la siguiente manera:

SW[0]	SW[1]	LED[0]	LED[1]	LED[2]	LED[6]	LED[7]
0	0	0	0	0	0	0
0	1	0	1	1	0	1
1	0	0	1	1	1	0
1	1	1	1	0	1	1

A partir del código que hemos implementado observamos su funcionamiento comprobando cada uno de los estados de los leds al activar/desactivar los dip switch SW[0] y SW[1].

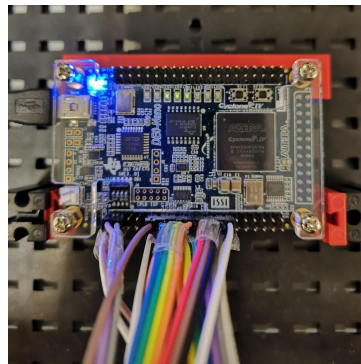
Si los resultados los obtenemos con nuestra tabla de verdad sabemos que el proyecto se ha realizado correctamente.

2.4. Funcionamiento del proyecto

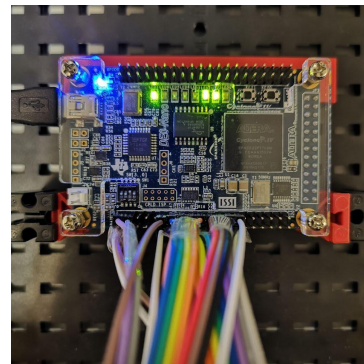
A partir de nuestra tabla de la verdad podemos observar el comportamiento de los leds dependiendo de los switches que estén activados.

Llamaremos *fase 0* cuando SW[0] y SW[1] estén desactivados, *fase 1* cuando SW[0] esté desactivado y SW[1] esté activado, *fase 2* cuando SW[0] esté activado y SW[1] esté desactivado y por último *fase 3* cuando SW[0] y SW[1] estén activados.

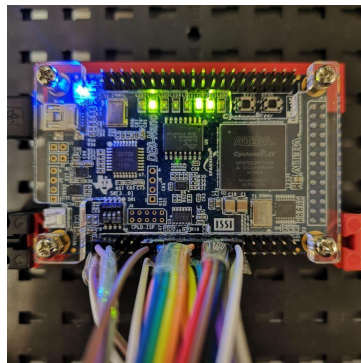
A continuación, adjuntamos las imágenes de la placa.



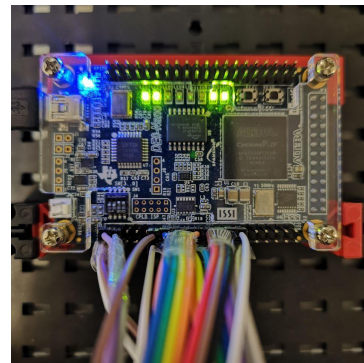
(a) Fase 0



(b) Fase 1



(c) Fase 2



(d) Fase 3

Figura 8: Fases del proyecto

2.5. Simulación del proyecto

En este apartado mostraremos la simulación del proyecto. Tras seguir los pasos explicados en el apartado 1 para crear el archivo obtenemos el siguiente cronograma:

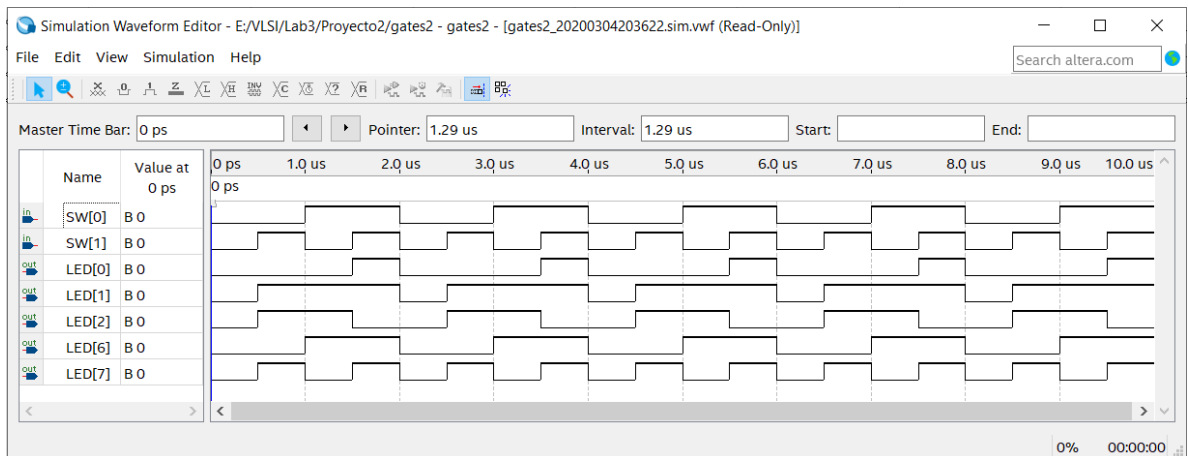


Figura 9: Simulación del circuito

2.6. Errores producidos en la realización del proyecto

En la realización del proyecto nos ha surgido un error al realizar la simulación. Este error lo hemos resuelto cambiando la configuración de la simulación desde la pestaña *Assignments* como se indica en la siguiente página: <https://forums.intel.com/s/question/0D50P00003yyGTISA2/error-when-using-university-program-vwf?language=es>