



# Universidad Nacional Autónoma de México

## Facultad de Ingeniería



Arquitectura de Computadoras 2020-1

Práctica 2: Introducción al lenguaje VHDL

Aguilar Enriquez Paul Sebastian

Berdejo Arvizu Oscar

Pérez Bueno Ana Laura

Rojas Héctor

Grupo: 01

23 de Septiembre del 2019

M.I. Jose Antonio de Jesus Arredondo Garza

# Objetivos

- Familiarizar al alumno en el conocimiento de los algoritmos de las máquinas de estados utilizando el lenguaje VHDL.

## Introducción

### Lenguaje de descripción de hardware

Un lenguaje de descripción de hardware (HDL, hardware description language) es un lenguaje de programación especializado que se utiliza para definir la estructura, diseño y operación de circuitos electrónicos, y más comúnmente, de circuitos electrónicos digitales, como el convertidor analógico-digital o cualquier antena satelital. Así, los lenguajes de descripción de hardware hacen posible una descripción formal de un circuito electrónico, y posibilitan su análisis automático y su simulación.

Los lenguajes de descripción de hardware se parecen mucho a otros lenguajes de programación de ordenadores tales como el C o Java: básicamente consisten en una descripción textual con expresiones, declaraciones y estructuras de control. Sin embargo, una importante diferencia entre los HDL y otros lenguajes de programación está en que el HDL incluye explícitamente la noción de tiempo.

### Máquinas de estados

Se denomina máquina de estados a un modelo de comportamiento de un sistema con entradas y salidas en donde las salidas dependen no solo de las señales de entradas actuales, sino también de las anteriores.

Las máquinas de estados se definen como un conjunto de estados que sirven de intermediarios en esta relación de entradas y salidas, haciendo que el historial de señales de entrada determine, para cada instante, un estado para la máquina de forma tal que la salida depende únicamente del estado y las entradas actuales.

Una máquina de estados se denomina máquina de estados finitos (FSM por finite state machine) si el conjunto de estados de la máquina es finito y es el único tipo de máquinas de estados que podemos modelar en un computador en la actualidad. Debido a esto se suelen utilizar los términos «máquina de estados» y «máquina de estados finitos» de forma intercambiable.

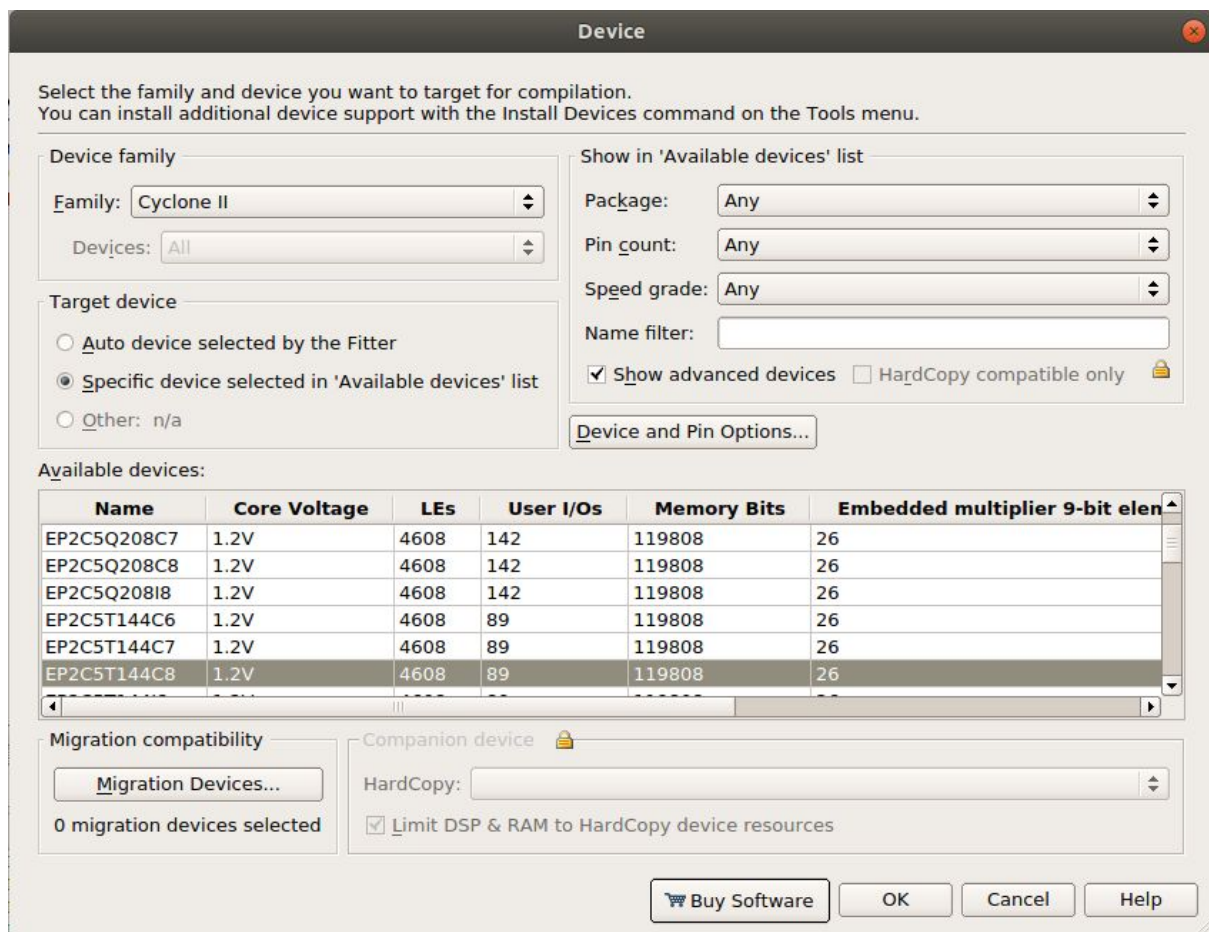
La representación de una máquina de estados se realiza mediante un diagrama de estados. Sin embargo también es posible utilizar un diagrama de flujo.

Es posible clasificar las máquinas de estados en aceptoras o transductoras:

- **Aceptoras** (también llamadas reconocedoras o discriminadoras). Son aquellas en donde la salida es binaria, depende únicamente del estado y existe un estado inicial. Puede decirse, entonces, que cuando la máquina produce una salida positiva es porque ha reconocido o aceptado la secuencia de entrada. En las máquinas de estados aceptoras, los estados con salida positiva se denominan estados finales.
- **Transductoras**. Son las más generales. Convierten una secuencia de señales de entrada en una secuencia de salida, pudiendo esta ser binaria o más compleja, según la entrada actual (no solo del estado) y pudiendo también prescindir de un estado inicial.

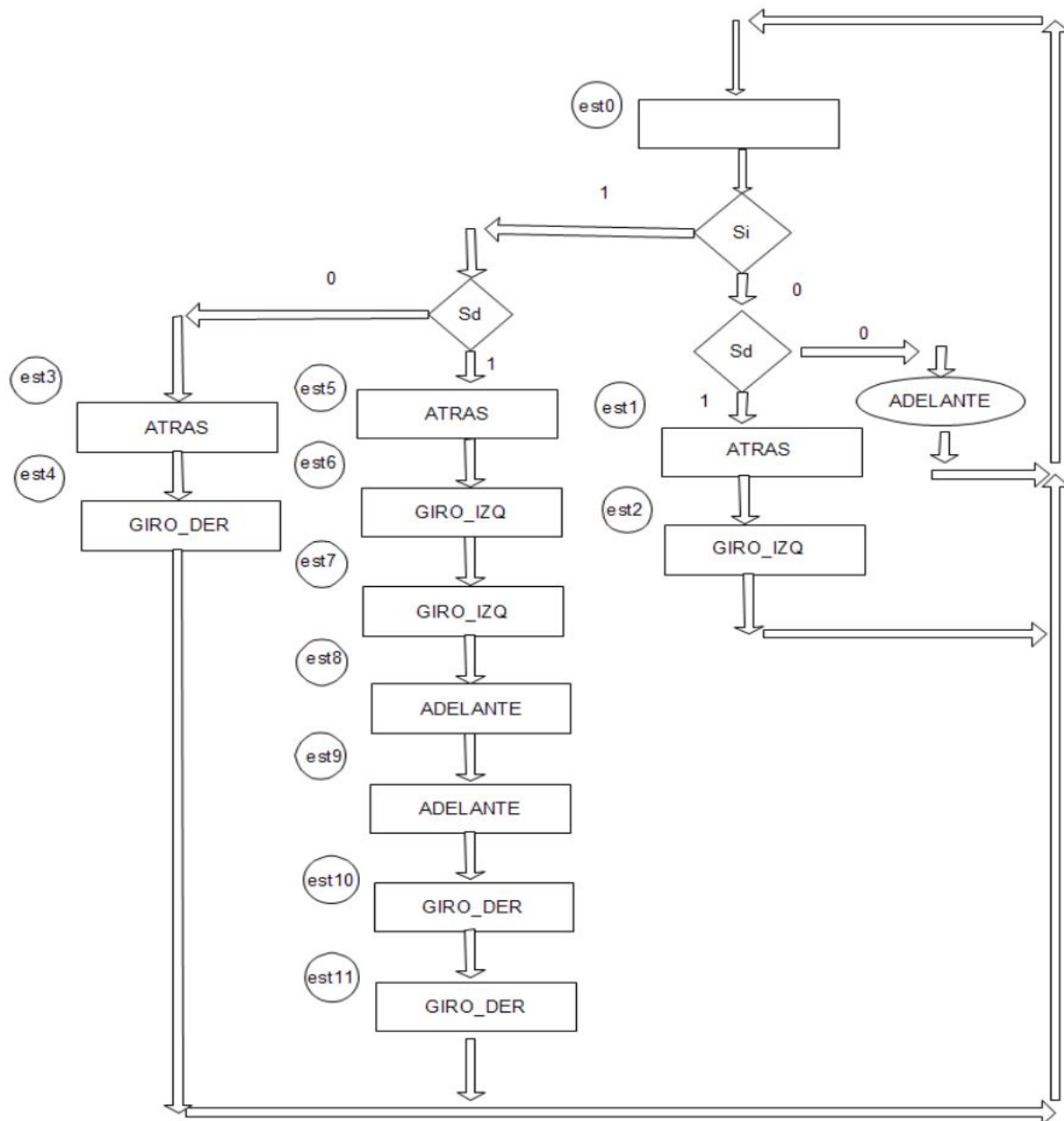
## Desarrollo

El proyecto se desarrolló para una tarjeta **Cyclone II** modelo **EP2C5T144C8**, por lo cual se utilizó Quartus 13.1.



**Figura 1:** Configuración del proyecto y la tarjeta.

Para el primer apartado se implementó el diseño del circuito electrónico de la siguiente carta ASM.

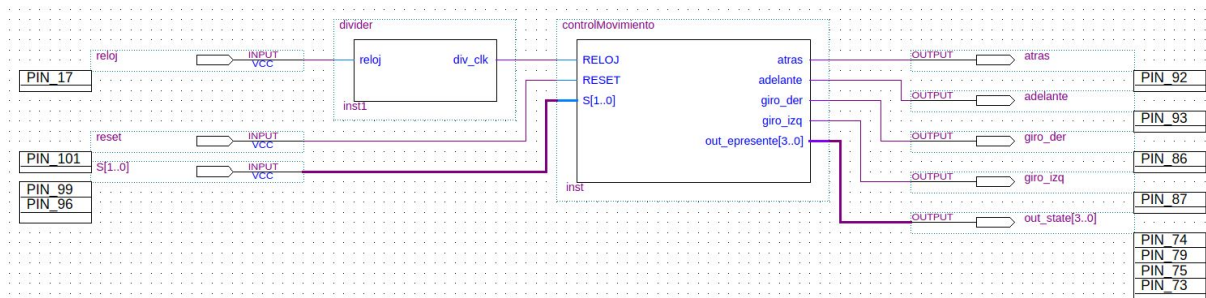


**Figura 2:** Algoritmo a implementar usando notación de carta ASM.

Para parte de la implementación se utilizó el código proporcionado en la práctica. El algoritmo anterior (representado mediante código VHDL) es una máquina de estados que funciona como algoritmo de control de movimientos-desplazamientos de un vehículo/robot.

Para la implementación completa se utilizó el divisor de tiempo de la práctica 1, un botón de reset, dos botones de entrada, uno para selección de giro a la derecha y otro para selección de giro a la izquierda, cuatro leds para la representación de los movimientos (adelante, atras, giro a la derecha, giro a la izquierda) y cuatro leds para la representación de los estados por los que va pasando la máquina de estados.

Al utilizar una tarjeta distinta a la propuesta por la práctica, las entradas y salidas no requieren una compuerta inversora. La implementación en diagrama de bloques quedó de la siguiente manera:



**Figura 3:** Implementación por bloques del algoritmo anterior con asignación de pines.

La asignación de pines quedó de la siguiente manera:

- **Entradas**

- reloj => Pin 17 => Reloj interno, no requiere conexión física
- reset => pin 101 => Se activa en bajo
- izquierda => Pin 99
- derecha => Pin 96

- **Salidas**

- adelante => Pin 93
- atras => Pin 92
- izquierda => Pin 86
- derecha => Pin 87
- estado0 => Pin 73 => bls (bit menos significativo)
- estado1 => Pin 75
- estado2 => Pin 79
- estado3 => Pin 74 => bms (bit más significativo)

Pin Planner - /home/penserbjorne/Proyectos/clase-arquitectura-computadoras-2020-1/pac...

File Edit View Processing Tools Window Help Search altera.com

Top View  
Wire Bond

Cyclone II  
EP2C5T144C8

Named: \* Edit: Filter: Pins: all

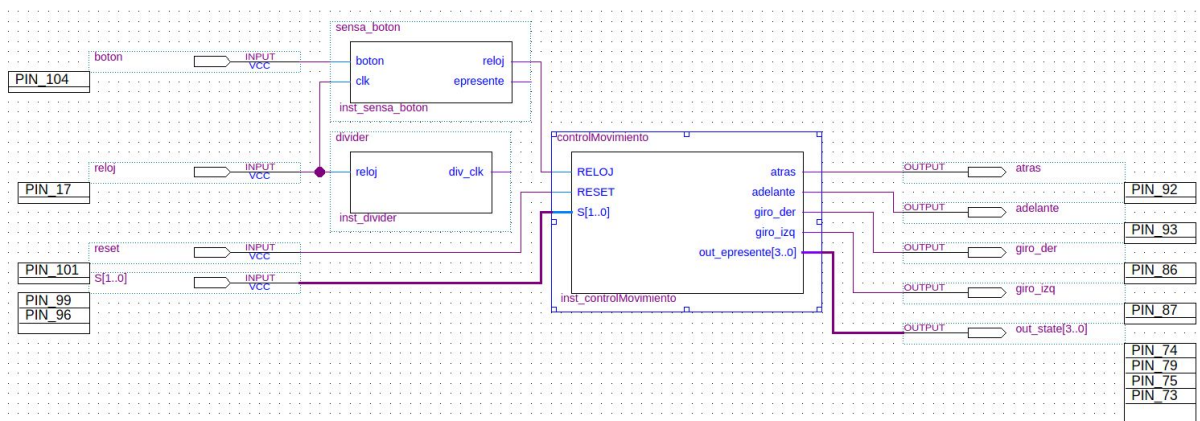
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location
adelante	Output	PIN 93	3	B3 N0	PIN 93
atras	Output	PIN 92	3	B3 N0	PIN 92
giro der	Output	PIN 86	3	B3 N1	PIN 86
giro izq	Output	PIN 87	3	B3 N1	PIN 87
out state[3]	Output	PIN 74	3	B3 N1	PIN 74
out state[2]	Output	PIN 79	3	B3 N1	PIN 79
out state[1]	Output	PIN 75	3	B3 N1	PIN 75
out state[0]	Output	PIN 73	3	B3 N1	PIN 73
reloj	Input	PIN 17	1	B1 N0	PIN 17
reset	Input	PIN 101	3	B3 N0	PIN 101
S[1]	Input	PIN 99	3	B3 N0	PIN 99
S[0]	Input	PIN 96	3	B3 N0	PIN 96

0% 00:00:00

**Figura 4:** Asignación de los pines en la sección de PINOUT de Quartus.

Se añadió otro elemento correspondiente a una máquina de estados que a partir de una señal de reloj externa de alta frecuencia y la de un botón de entrada sirve para sensar cuando este es oprimido y liberado, enviando una señal que permite ver el desempeño de otra máquina de estados conectado a el paso a paso. Básicamente, un botón que en función de un reloj y un botón permite avanzar otra máquina de estado paso a paso.

El código de la máquina de estados fue proporcionado por la practica y se implementó como otro bloque llamado sensa boton el cual se conecta al reloj del circuito y a un botón de entrada. La implementación del circuito es la siguiente:



**Figura 5:** Implementación por bloques con el módulo de sensado boton con asignación de pines.

La asignación de pines quedó de la siguiente manera:

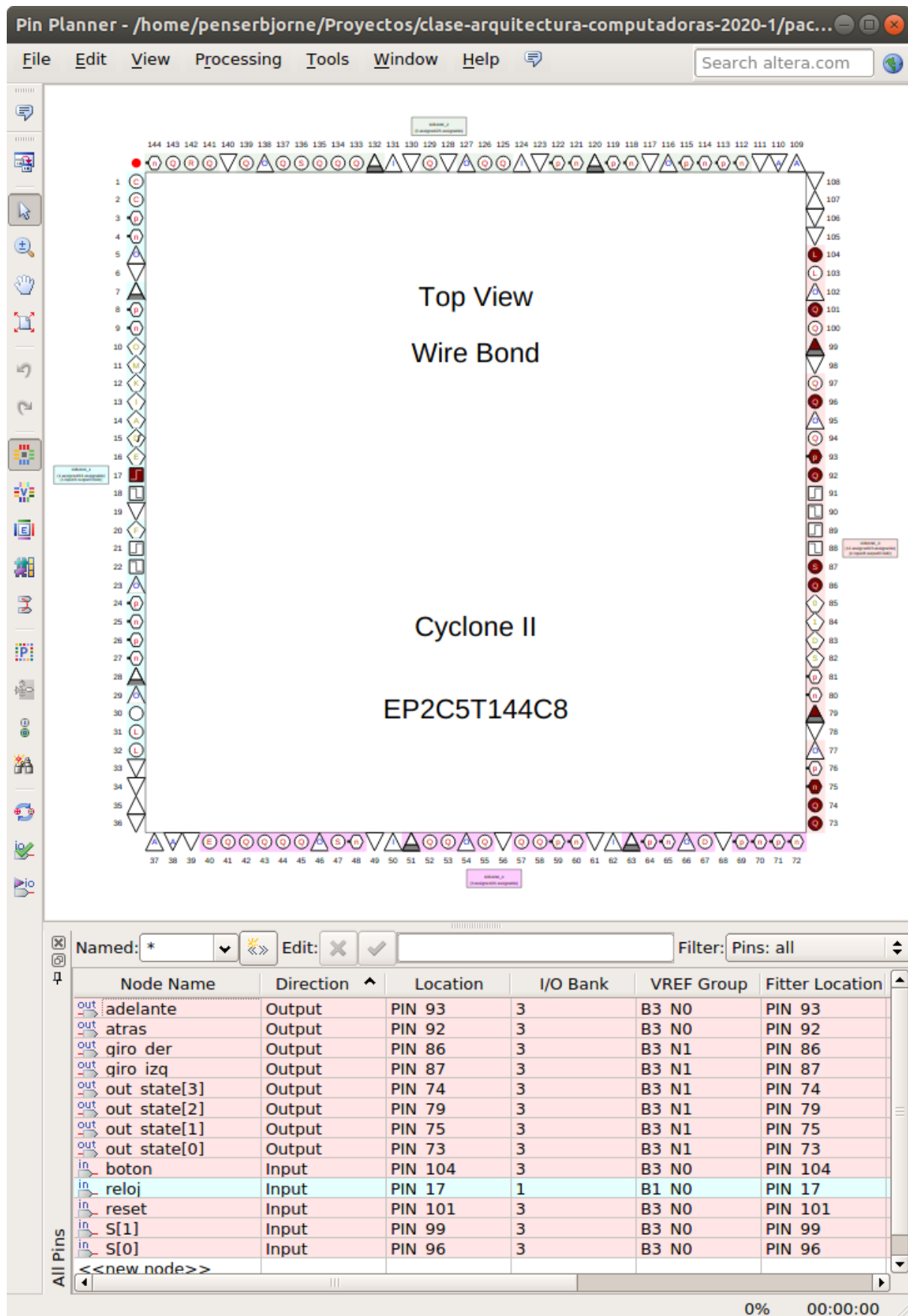
- **Entradas**

- o reloj => Pin 17 => Reloj interno, no requiere conexión física
- o reset => pin 101 => Se activa en bajo
- o izquierda => Pin 99
- o derecha => Pin 96
- o boton => Pin 104

- **Salidas**

- o adelante => Pin 93
- o atras => Pin 92
- o izquierda => Pin 86
- o derecha => Pin 87
- o estado0 => Pin 73 => bls (bit menos significativo)
- o estado1 => Pin 75
- o estado2 => Pin 79
- o estado3 => Pin 74 => bms (bit más significativo)





**Figura 6:** Asignación de los pines en la sección de PINOUT de Quartus.



## Conclusiones

Como pudimos ver en la primera práctica, los lenguajes de descripción de hardware junto a los FPGA son una excelente alternativa para el diseño e implementación de circuitos (sobre todo de circuitos digitales), y ahora en la segunda práctica podemos concluir que de igual manera son una excelente alternativa para la implementación de algoritmos y máquinas de estados, sobre todo aquellos que pueden simplificarse y reducirse a circuitos, lo cual empata con el objetivo de los lenguajes de descripción de hardware.

En cuanto a la práctica, esta sirvió como tema introductorio al acercamiento de los lenguajes de descripción de hardware y los FPGA. La práctica fue sencilla por lo que seguir las instrucciones fue suficiente para poder realizarla sin mayor complicación.

Consideramos de gran utilidad estas tecnologías ya que aplicándolas a la ingeniería permiten extender y aplicar conocimiento de manera sencilla como es la implementación de máquinas de estado o algoritmos que requieren elementos de hardware, como compuertas.

## Referencias

- [https://es.wikipedia.org/wiki/Lenguaje\\_de\\_descripci%C3%B3n\\_de\\_hardware](https://es.wikipedia.org/wiki/Lenguaje_de_descripci%C3%B3n_de_hardware)
- [https://es.wikipedia.org/wiki/M%C3%A1quina\\_de\\_estados](https://es.wikipedia.org/wiki/M%C3%A1quina_de_estados)