



Escola Universitària  
Politécnica de Mataró

**Ingeniería Técnica Industrial: Especialidad Electrónica Industrial**

**Tarjeta de evaluación y prácticas para dispositivos CPLD**

**Autor: David Ferrés**  
**Profesor ponente: Joan Gil**  
Otoño 2008



## **Resumen**

Con este proyecto se pretende crear una tarjeta de evaluación para prácticas, implementando un dispositivo lógico programable complejo (CPLD). Esta tarjeta tiene como finalidad ayudar al alumno a comprender y comprobar sus diseños digitales mediante el uso de dispositivos programables.

Es evidente que la utilización de este tipo de tarjetas, hace posible la comprensión y asimilación de los conocimientos teóricos en el diseño de CPLDs. Haciendo viable que, de la simulación de las prácticas se pueda pasar a la comprobación real, de una forma fácil y sin tener que realizar un prototipo para alcanzar este fin.

Este proyecto engloba tres objetivos:

Una primera visión general de los dispositivos PLDs, su capacidad y utilidad, donde se expone de forma breve su arquitectura, funcionamiento, etc.

La parte principal y básica del proyecto, esta formada por el diseño y la descripción de una tarjeta de evaluación, para la realización de prácticas con estos dispositivos.

Finalmente y como complemento al proyecto. Se plantean algunos ejercicios a modo de ejemplo para la realización de prácticas con esta tarjeta.



## **Resum**

Amb aquest projecte es pretén crear una targeta d'avaluació per pràctiques, implementant un dispositiu lògic programable complex (CPLD). Aquesta targeta té com a finalitat ajudar l'alumne a comprendre i comprovar els seus dissenys digitals mitjançant l'ús de dispositius programables.

És evident que la utilització d'aquest tipus de targetes fa possible la comprensió i assimilació dels coneixements teòrics en el disseny de CPLDs. Fent viable que, de la simulació de les pràctiques es pugui passar a la comprensió real, d'una forma fàcil i sense necessitat de realitzar un prototip per aconseguir aquest fi.

Aquest projecte engloba tres objectius:

Una primera visió general dels dispositius PLDs, la seva capacitat i utilitat, on s'exposa de forma breu la seva arquitectura, funcionament, etc.

La part principal i bàsica del projecte està formada pel disseny i descripció d'una targeta d'avaluació, per la realització de pràctiques amb aquests dispositius.

Finalment, i com complement al projecte, es plantegen alguns exercicis a mode d'exemple per a la realització de pràctiques amb aquesta targeta.



## **Abstract :**

The aim of this project is to get an electronic circuit used in practical evaluation implementing a logical device with complex programming (CPLD).

This electronic circuit has an objective : to be a help when the pupil wants to realize and to check digital designs with programming devices.

Logically the use of this kind of electronic circuits improve the comprehension and understood about theoretic knowledge in the design of CPLDs. Doing reliable to compare the practice simulation with the real result , easily and without having to make a prototype in order to achieve this result.

This project has three objectives:

A general and first vision about PLDs devices, its capability and use where everybody can see its architecture , resolution work and so on...

The principal and basic point of this project include the design and description of the electronic circuit in order to get the fulfilment of practices with these devices.

Finally, being a support about the project we arrange some exercises with examples for practical use about this electronic circuit.



## Índice

1	OBJETIVOS.....	3
1.1	Dificultades.....	3
1.2	El Proyecto: Entrenador para CPLD .....	4
2	TEORÍA DE LOS PLDS.....	5
2.1	Introducción a los PLDs .....	5
2.2	Dispositivos lógicos programables simples (SPLDs).....	6
2.2.1	PROM.....	7
2.2.2	PAL.....	8
2.2.3	FPLA .....	9
2.3	Dispositivos Lógicos Programables Complejos (CPLDs).....	10
2.3.1	Arquitectura de los CPLDs.....	11
2.3.2	Matriz de conexión global .....	11
2.3.3	Bloque lógico y Macrocelas .....	12
2.3.4	Bloque de entrada/salida.....	12
2.4	Conjuntos configurables de puertas (FPGA).....	12
2.4.1	Bloques lógicos internos.....	13
2.4.2	Los Bloques lógicos de E/S.....	13
2.4.3	Recursos de interconexión o líneas de conexión.....	14
2.5	Híbridos .....	14
3	ARQUITECTURA DE LOS CPLDS DE ALTERA.....	15
3.1	Historia de ALTERA.....	15
3.2	Dispositivos Lógicos programables de la familia MAX.....	15
3.2.1	Bloques lógicos o LABs.....	16
3.2.2	Bloques de entrada/salida .....	18
3.2.3	Matriz de interconexión programable (PIA) .....	19
4	DISEÑO Y COMPONENTES .....	21
4.1	Diseño y componentes que confeccionan el entrenador.....	21
4.2	Características del CPLD .....	24
4.3	Tabla resumen de las conexiones de entrada/salida del CPLD .....	25
4.4	El Estándar JTAG.....	29
4.4.1	EL STD.IEEE-1149.1 .....	29
4.5	Estructura del bus JTAG .....	30
4.5.1	Entrada de datos .....	30
4.5.2	Salida de datos de prueba .....	30
4.5.3	Selección del modo test.....	31
4.5.4	Reloj .....	31
4.6	Salidas del entrenador.....	32
4.6.1	<i>Displays</i> de 7 segmentos .....	32
4.6.2	Conjunto de resistencias .....	33
4.6.3	Transistores PNP .....	33
4.6.4	Leds .....	33
4.6.5	Pantalla LCD .....	35
4.6.6	Pins de control del <i>Display</i> LCD.....	35
4.7	Entradas del entrenador .....	37
4.7.1	Teclado matricial.....	37
4.7.2	Pulsadores.....	40
4.7.3	Interruptores tipo DIL o Microinterruptores .....	40

## 2 -Objetivos

4.7.4	Conectividad de los diferentes módulos que conforman el entrenador .....	41
4.7.5	Conexión de los diferentes módulos .....	41
4.7.6	Disposición de los relojes.....	42
4.7.7	Oscilador de cuarzo.....	42
4.7.8	Oscilador 555 .....	43
5	MANUAL DE CREACIÓN Y FUNCIONAMIENTO .....	47
5.1	Pasos a seguir para crear la PCB.....	47
5.2	Funcionamiento del entrenador.....	49
6	PRÁCTICAS.....	51
6.1	Práctica 1: Visualización dinámica .....	51
6.1.1	Objetivo.....	51
6.1.2	Enunciado.....	51
6.1.3	Hardware necesario .....	51
6.2	Práctica 2: Contador programable.....	52
6.2.1	Objetivo.....	52
6.2.2	Enunciado.....	52
6.3	Práctica 3: Control del teclado matricial.....	53
6.3.1	Objetivo.....	53
6.3.2	Enunciado.....	53
6.3.3	Módulos necesarios para la creación del teclado matricial.....	53
6.3.4	Generador de Señales.....	53
6.3.5	Contador.....	55
6.3.6	Descodificador .....	56
6.3.7	Control.....	57
7	MATERIALES DE FABRICACIÓN.....	59
7.1	Listado de componentes que conforman el entrenador.....	59
8	CONCLUSIÓN.....	61
9	PRESUPUESTO .....	63
9.1	Costes del material .....	63
9.2	Costes relacionados con la ingeniería .....	65
9.3	Amortización del instrumental .....	65
9.4	Coste de fabricación del prototipo .....	65
9.5	Coste de fabricación.....	65
9.6	Estudio económico para la fabricación de 100 Uds.....	66
9.7	Estudio del precio de venta en el mercado.....	66
10	ESQUEMAS .....	67
10.1	Esquemas de conexionado de los módulos .....	67
11	ÍNDICE DE FIGURAS.....	71
12	ÍNDICE DE TABLAS .....	73
13	BIBLIOGRAFÍA.....	75

# **1 OBJETIVOS**

El objetivo es el de desarrollar un entrenador de CPLDs de ALTERA para ser utilizado posteriormente por los alumnos a la hora de realizar prácticas.

La finalidad del proyecto es la de suministrar una herramienta útil para que el usuario pueda poner en practica los conocimientos adquiridos sobre los dispositivos lógicos programables.

La tarjeta de evaluación ha sido dotada de elementos que se ha creído que favorecerán el estudio de los programas ejecutados en el CPLD como pueden ser: Leds, pulsadores, elementos de visualización...

Con el objetivo de ayudar al alumno en la utilización de la tarjeta de evaluación se ha querido distribuir este proyecto en tres partes, una primera parte que explica el funcionamiento de los CPLDs, otra parte que explica los elementos que conforman el entrenador, y por ultimo las practicas que tienen como objetivo ayudar a familiarizarse y comprender el funcionamiento de la tarjeta.

## **1.1 Dificultades**

Normalmente el problema que tiene el alumno a la hora de realizar un programa nuevo para un CPLD, es que necesita comprobar los resultados de alguna manera antes de dar el visto bueno a dicho programa. Se da pues la necesidad de fabricar la aplicación a la que va destinada el programa con las complicaciones que ello conlleva a la hora de fabricarlo: resolviendo los problemas que surjan sobre la marcha, teniendo que reconfigurar el modelo inicial si este no responde como debería, o bien estudiando complicados cronogramas que representan la respuesta del modelo creado para el CPLD.

#### 4 -Objetivos

Por ello, la tarjeta de evaluación viene a suplir las dificultades que surgen a la hora de implementar el programa para verificar el resultado.

### **1.2 El Proyecto: Entrenador para CPLD**

Es un entrenador para aplicaciones basadas en el dispositivo CPLD.

Este entrenador se ha diseñado para que el usuario pueda grabar en el chip de Altera los proyectos creados, y mediante una configuración personalizada (a las necesidades de los proyectos) de los diferentes módulos que conforman el entrenador, estudiar las diferentes respuestas que este nos dé, para así poder realizar las modificaciones que se crean oportunas, e incluso estudiar su viabilidad en un entorno más o menos cercano a lo que sería la realidad.

Con este entrenador también se puede ahorrar tiempo a la hora de verificar la respuesta del sistema a los cambios realizados en el programa inicial, debido a que el CPLD es programable en el sistema (*ISP, In-System Programmability*).

## **2 TEORÍA DE LOS PLDS**

### **2.1 Introducción a los PLDs**

Aunque existen en el mercado muchos dispositivos lógicos programables (PLDs) con arquitecturas diferentes, todos están basados en la siguiente idea estructural: están constituidos por bloques lógicos configurables, conectados mediante bloques de conexión programables.

Esta organización, tiene sus antecedentes en las matrices lógicas programables y suele recibir el nombre de matricial. Los circuitos PLD tuvieron un gran desarrollo desde que a mediados de la década de los 70 se realizaron los primeros secuenciadores lógicos programables o PLS.

Su utilización provoca que se mejoren aspectos tales como la reducción del número de elementos necesarios, la disminución del tiempo necesario para crear el diseño, el aumento de la respuesta y la disminución de conexiones externas lo que provoca la disminución a la hora de generar ruido. Gracias a estas mejoras obtenemos una disminución en los costes de producción y un aumento de fiabilidad del sistema, cosa que hace que los PLDs sean una buena elección a la hora de empezar a diseñar el proyecto.

Los PLDs presentes en el mercado tienen diferentes arquitecturas dependiendo del fabricante, incluso hay algunos que tienen arquitecturas híbridas, por ello se pueden organizar según su capacidad de respuesta, de su arquitectura, etc...

La clasificación de los PLDs es la siguiente:

- Dispositivos lógicos programables simples (SPLD)
- Dispositivos lógicos programables complejos (CPLD)
- Matrices de puertas programables por campo (FPGA)
- Dispositivos Híbridos (CPLD-FPGA)

## 2.2 Dispositivos lógicos programables simples (SPLDs)

Se les denomina así a todos los PLDs que están formados por bloques lógicos programables conectados a macroceldas con interconexiones programables y de un tamaño del orden de 22 macroceldas.

El diagrama típico de bloques de un SPLD está formado por un bloque lógico y las macroceldas.

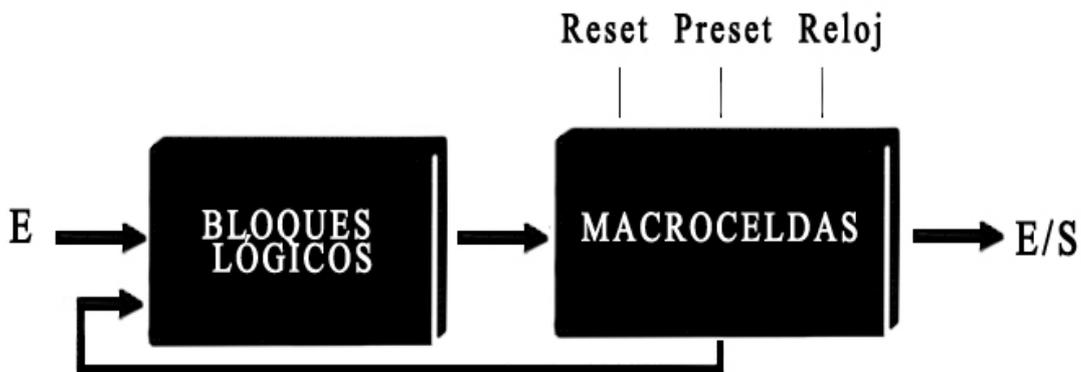


Figura 1: Diagrama de bloques de un SPLD

Un bloque lógico no es más que las entradas conectadas a la matriz de entrada (o matriz AND) y las salidas conectadas a la matriz de salida (matriz OR)



Figura 2: Composición de un bloque lógico

Dependiendo de la configuración que se puedan aplicar a las matrices de entrada/salida del bloque lógico existen los siguientes tipos de SPLDs:

- PROM *Programmable Read Only Memory*
- PAL *Programable Array Logic*
- FPLA o PLA *Field Programmable Logic Array*

A continuación se explica la funcionalidad de los bloques lógicos de los SPLDs que tienen estas arquitecturas.

### **2.2.1 PROM**

Son memorias de lecturas programadas por el usuario, según la tecnología de borrado o la técnica utilizada para ello se puede distinguir entre: PROM, OTPROM, UVPROM, EEPROM, EEPROM FLASH.

Los bloques lógicos formados con esta tecnología tienen la matriz de entrada programada de fábrica mientras que la matriz de salida debe ser programada por el usuario y los SPLDs no dispondrán de macroceldas. Ya que con “n” entradas tendremos “ $2^n$ ” salidas

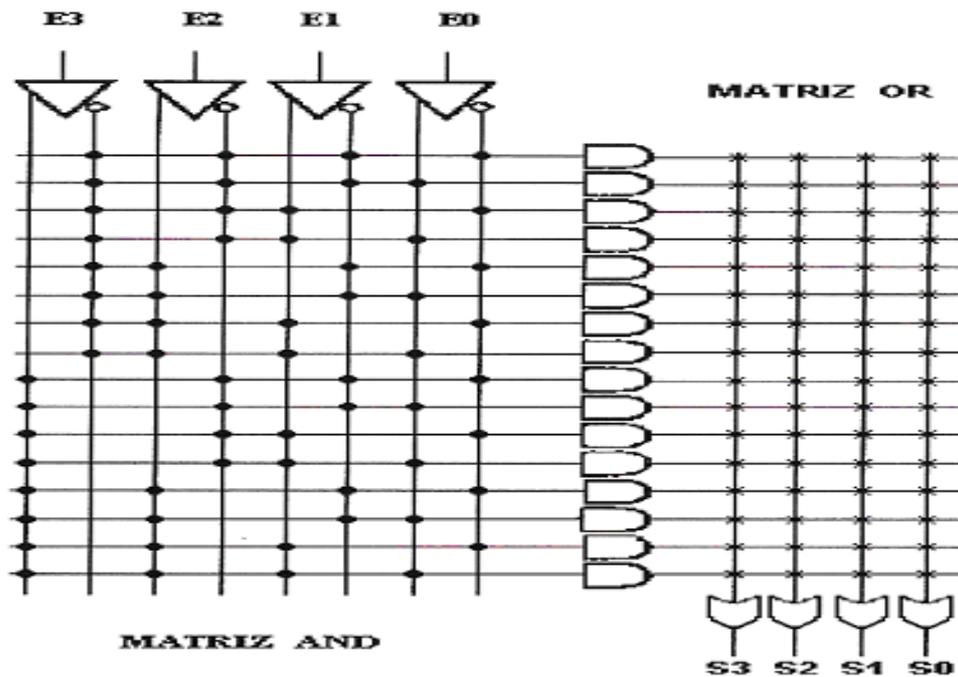


Figura 3: Diagrama de programación del bloque lógico de una PROM

## 2.2.2 PAL

Los SPLDs que tienen esta arquitectura como bloque lógico a diferencia de la arquitectura PROM necesitarán de macroceldas; esta arquitectura es la más extendida entre los fabricantes de SPLDs.

En una PAL la matriz de entrada AND es la programable por el usuario, en cambio, la matriz de salida OR viene programada de fábrica de una manera curiosa, ya que teniendo “n” entradas se dispone a la salida de menos de “ $2^n$ ” términos, por ello es necesario el bloque de las macroceldas para poder realimentar la señal y así tener mas términos a la salida.

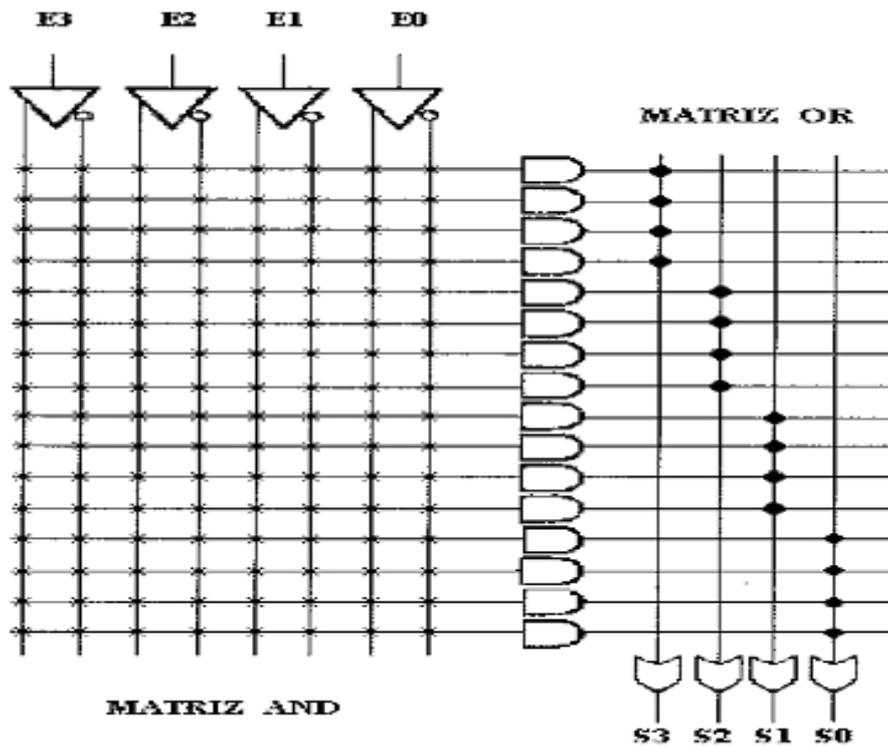


Figura 4: Diagrama de programación del bloque lógico de una PAL

### 2.2.3 FPLA

Este sistema de bloque lógico posee una estructura más flexible ya que pueden ser programadas ambas matrices por el usuario, tanto la de entrada como la de salida, por el contrario la velocidad será menor y se suele usar sólo como máquina de estados.

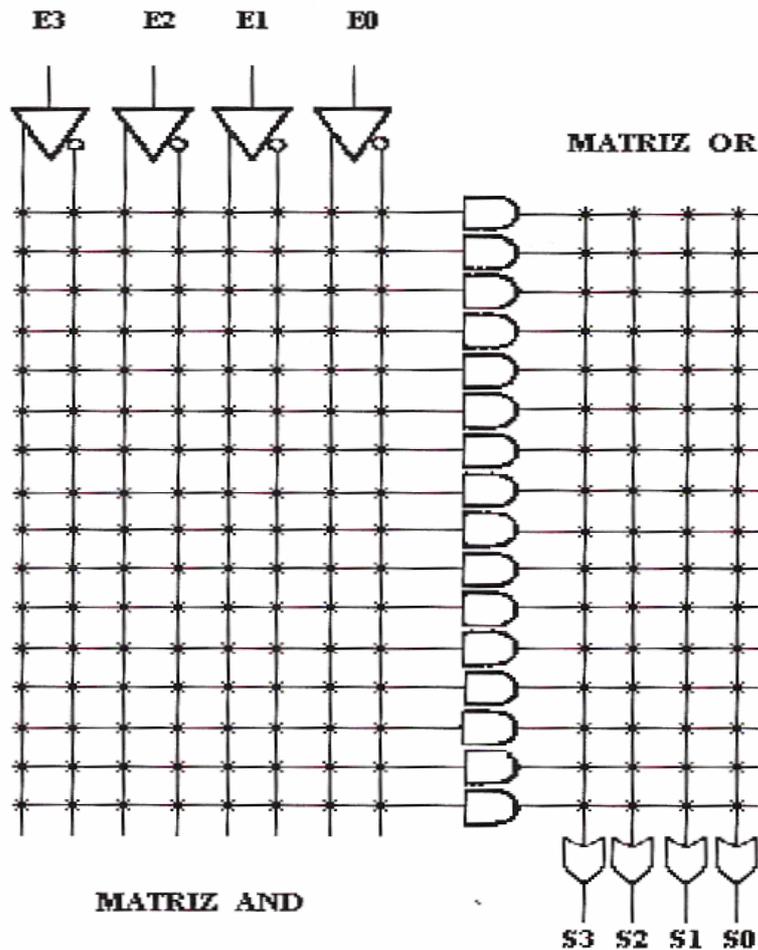


Figura 5: Diagrama de programación del bloque lógico de una FPLA

### 2.3 Dispositivos Lógicos Programables Complejos (CPLDs)

Los CPLDs se crearon para poder diseñar sistemas digitales muy complejos que los SPLDs debido a su simplicidad eran incapaces de resolver. Por ello los CPLDs se crearon con la idea de ser un conjunto de bloques de SPLDs interconectados mediante una matriz de conexión.

A raíz de su creación se ha logrado disminuir el coste de desarrollo y el tiempo de respuesta del sistema, además la vida del producto aumenta considerablemente gracias a la capacidad de actualización que tienen los CPLDs.

### 2.3.1 Arquitectura de los CPLDs

La mayoría de los CPLDs tienen la misma arquitectura interna, dividida en tres partes que son:

- Matriz de conexión global
- Bloques lógicos con sus macroceldas
- Bloques de entrada y salida.

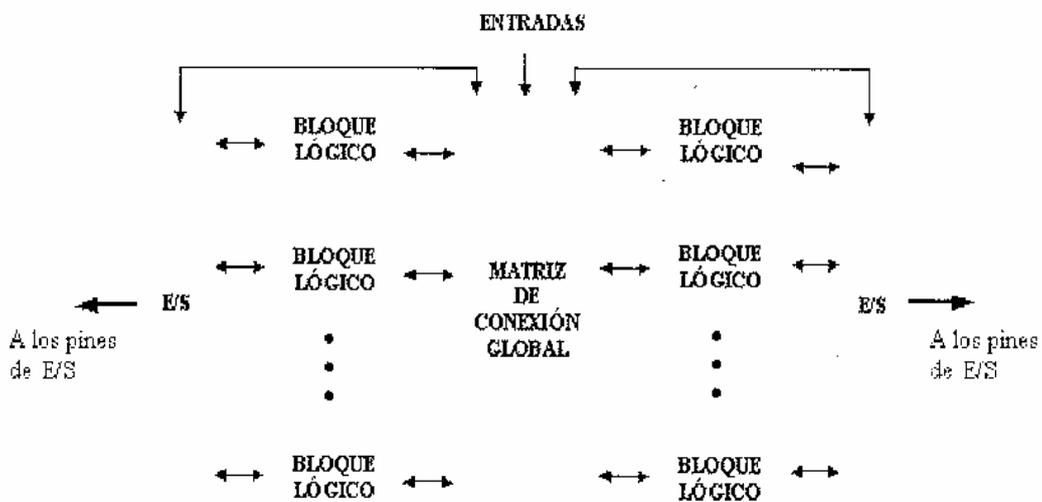


Figura 6: Ejemplo de la arquitectura interna de un CPLD

### 2.3.2 Matriz de conexión global

Es una matriz programable que permite realizar conexiones entre el bloque lógico y el bloque de entrada/salida.

Esta matriz conecta todo el sistema y se puede tener acceso a todos los pins del sistema, ya sean entradas/salidas, entradas dedicadas o realimentaciones procedentes de las macroceldas.

### **2.3.3 Bloque lógico y Macroceldas**

El bloque lógico de un CPLD tiene la estructura de un SPLD completo, por lo que se puede decir que un CPLD está compuesto por varios SPLDs

Las macroceldas son los registros básicos disponibles en los bloques lógicos. Junto con las señales de control pueden hacer funciones como máquinas de estado, contadores, etc...

Hay dos tipos de macroceldas: de entrada/salida y de realimentación.

Las macroceldas de entrada/salida están unidas directamente al bloque E/S.

Las macroceldas de realimentación no están conectadas al bloque de E/S, sino que están conectadas directamente a la matriz de conexión global.

En un CPLD la cantidad de macroceldas puede variar de 32 a más de 3000.

### **2.3.4 Bloque de entrada/salida**

Este bloque es el encargado de conectar los pins del dispositivo a las macroceldas del bloque lógico. Configurándose como pins de entrada, salida o de alta impedancia.

El número de E/S puede variar hasta más de 200 dependiendo del encapsulado.

## **2.4 Conjuntos configurables de puertas (FPGA)**

Los FPGAs están formados por un mar de bloques lógicos rodeados por una matriz de bloques de entrada/salida interconectados por líneas programables.

Un FPGA contiene desde 64 a más de 1000 bloques lógicos, al ser tan numerosos estos bloques, no están interconectados entre si, sino que se conectan dependiendo de las necesidades del usuario, usando programas de diseño como si fuera el *autorouter* que utilizamos a la hora de crear una PCB.

Las FPGA derivan de los circuitos integrados creados a semimedida.

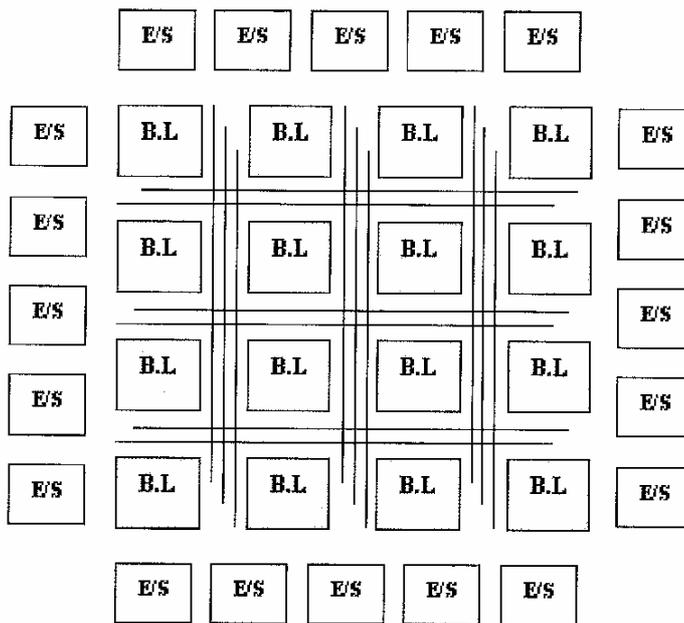


Figura 7: Diagrama de bloques de un FPGA

### 2.4.1 Bloques lógicos internos

Son recursos que permiten realizar diversas funciones lógicas, su complejidad varía desde un par de transistores hasta un conjunto de memorias de acceso aleatorio denominadas tablas de consultas.

En el primer caso se obtienen gran cantidad de bloques lógicos muy sencillos con capacidad de realizar un número muy limitado de funciones lógicas por cada uno de los bloques. Con el segundo método se dan menos puertas lógicas pero una capacidad para realizar funciones lógicas más elevadas.

### 2.4.2 Los Bloques lógicos de E/S

Son recursos que establecen el enlace entre los bloques lógicos internos y los terminales de entrada/salida. Al igual que en el caso anterior estos bloques pueden ser más o menos complejos.

### **2.4.3 Recursos de interconexión o líneas de conexión**

Son un conjunto de líneas e interruptores programables que permiten transmitir las señales entre los bloques lógicos internos; y entre ellos y los bloques de entrada/salida, recibiendo en inglés la denominación de *Routing Channels*.

## **2.5 Híbridos**

Se denominan así aquellos dispositivos que mezclan arquitecturas diferentes, para hacer un dispositivo con características comunes a otros.

Pudiendo ser:

- FPGA-CPLD
- FPGA-ASIC (Circuitos de Aplicaciones Específicas)
- CPLD + Periféricos

### **3 ARQUITECTURA DE LOS CPLDS DE ALTERA**

#### **3.1 Historia de ALTERA**

La empresa Altera se fundó en 1983, lanzando su primer PLD al año siguiente el EP300. En 1988 Altera presentó su primer PLD avanzado a los que dio la denominación genérica de arquitectura MAX (acrónimo de *Multiple Array Matrix*). En 1989 creó un lenguaje propio denominado AHDL *Altera Hardware Description Language*. Dicho lenguaje permite la descripción de sistemas tanto en nivel de comportamiento como en el estructural. En 1990 constituyó el sistema de diseño denominado MAX-PLUS para Unix y un año después vería la luz MAX-PLUSII para Windows, permitiendo describir los sistemas digitales en VHDL y Verilog.

La segunda familia de PLDs avanzados fue la MAX7000 comercializada en 1991. Dos años más tarde salía a la venta la familia MAX7000E. En 1994 se podían obtener los primeros CPLD denominados MAX9000 y durante los años siguientes salió la segunda generación de PLDs avanzados de la familia MAX7000S. Altera también ha comercializado PLDs complejos como FLASHlogic.

#### **3.2 Dispositivos Lógicos programables de la familia MAX**

Los elementos que conforman un CPLD de Altera, y que todas las familias MAX tienen, son los siguientes:

- Un conjunto de bloques lógicos, denominado por Altera como *Logics Array Blocks* o LABs.
- Bloques de entrada/salida denominados *I/O Control Blocks*.
- Una matriz de interconexión denominada por Altera como PIA, *Programmable Interconnect Array*.

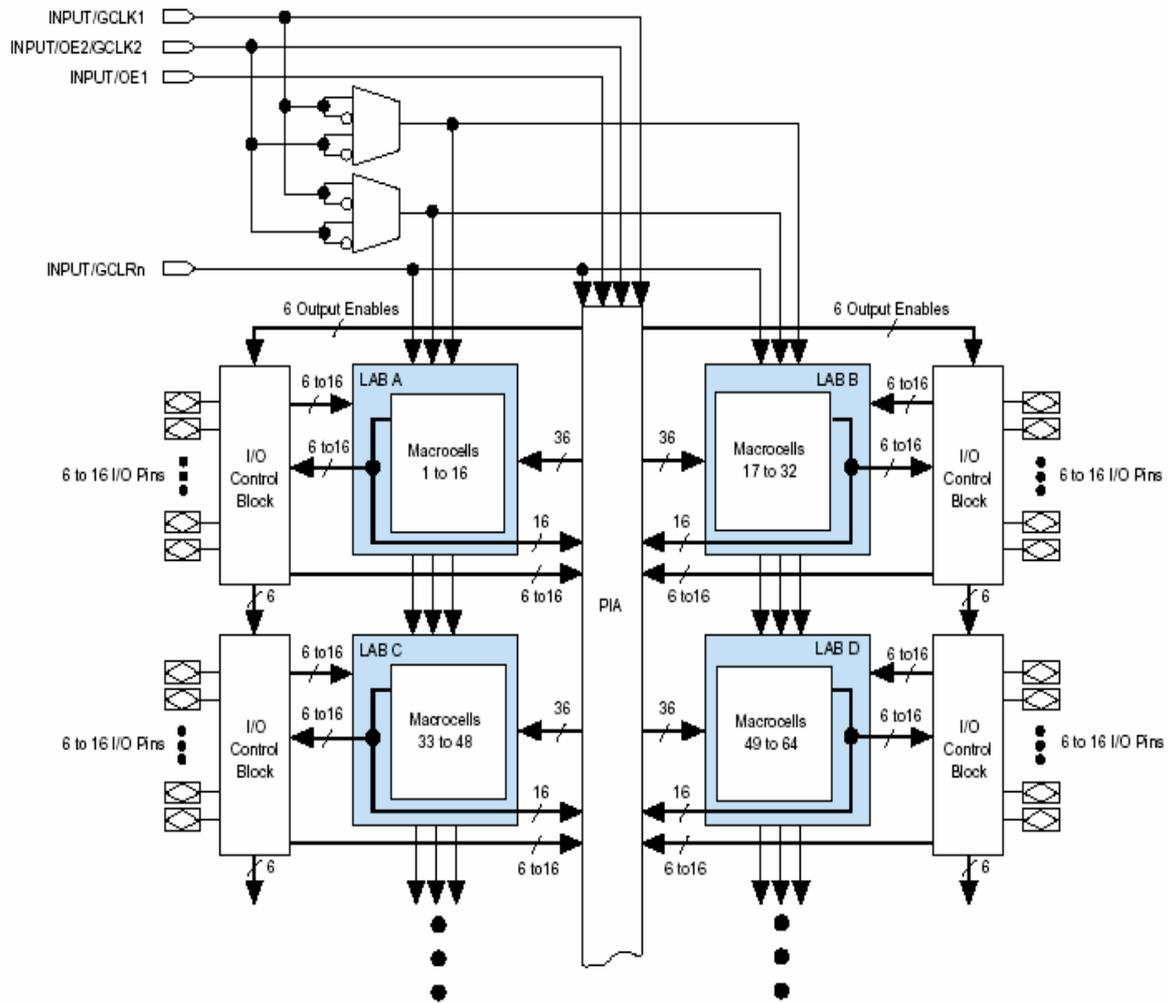


Figura 8: Diagrama de bloques de un CPLD de la familia MAX

Este es el típico diagrama de bloques de un CPLD de Altera perteneciente a la familia MAX7000S donde podemos ver las diferentes entradas de señales como la del reloj o la del *clear*, la de habilitación etc.

Además podemos ver como se interconectan los diferentes bloques entre ellos.

Pasaremos a describir cada uno de los bloques:

### 3.2.1 Bloques lógicos o LABs

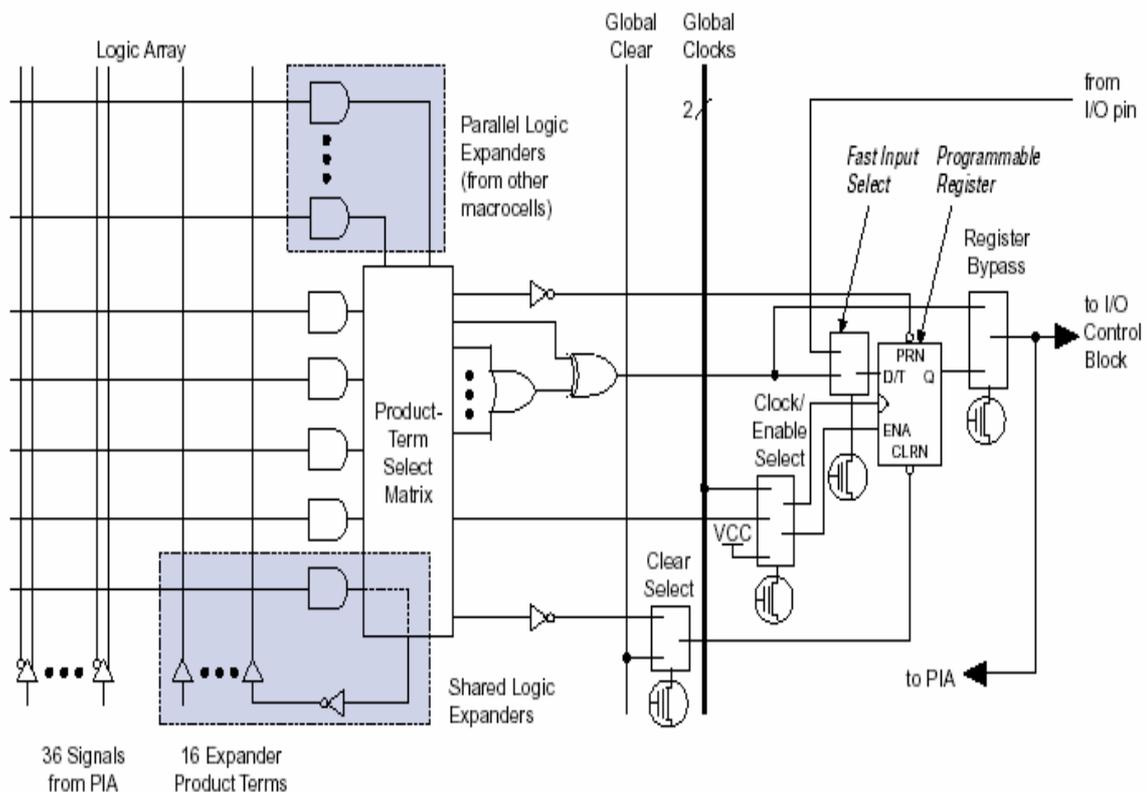
Podríamos decir que cada LAB constituye en si un SPLD formado por un conjunto de macroceldas que comparten una matriz de interconexión.

Estas macroceldas tienen unas características comunes a toda la familia MAX como pueden ser:

Tienen un único biestable y una sola realimentación. Pero si a la macrocelda se le añade el bloque de entrada/salida, el CPLD tendrá 2 realimentaciones. Si entramos en profundidad en lo que es una macrocelda típica que no forma parte de esta familia, veremos que sólo tenemos una entrada de realimentación, por lo que no es posible utilizar el terminal de entrada/salida y al mismo tiempo el biestable como memorización del estado interno. Lo que hacen otras familias es poner más realimentaciones a las macroceldas para así solucionar esta limitación.

El biestable se puede programar para que funcione como una bascula D, T, JK o SR.

La entrada del reloj biestable se puede obtener a través de una señal de reloj global o mediante la matriz de conexión.



**Figura 9: Diagrama de funcionamiento de una macrocelda**

En este diagrama de bloques se puede ver como es una macrocelda perteneciente a un bloque lógico de un CPLD de la familia MAX7000S

Sin entrar en detalle podemos ver que la macrocelda dispone de una matriz de selección de productos (*Product Term Select Matrix*) de ella salen los términos que se envían a las puertas OR y XOR, además de las señales de control, como el *Clear*, *Preset Clock* y *Clock Enable*.

También tenemos dos entradas denominadas de expansión cuya finalidad es la de realizar operaciones lógicas complejas empleando poco tiempo.

La diferencia entre las macroceldas de la familia Max7000S y las demás familias, es que al principio del registro hay un multiplexor que nos permite seleccionar la señal que proviene de la puerta O-Exclusiva o la que proviene del terminal Entrada/Salida, para sincronizar una variable de entrada sin tener que pasar por la matriz de conexión global o local, sólo para evitar retardos.

### **3.2.2 Bloques de entrada/salida**

Estos bloques también denominados por Altera como *I/O Control Block* permiten configurar de forma individual los terminales como entradas, salidas o bidireccionales.

Cada bloque dispone de ocho salidas que mediante un multiplexor y dependiendo de cómo se conecten funcionan de una manera o de otra. El multiplexor se puede conectar a VCC funcionando como salida, o a GND haciendo que el bloque funcione como entrada. También se puede conectar a una de las seis señales de habilitación, estas señales lo que hacen es habilitar un grupo de pins para que funcionen de manera independiente, por ejemplo podemos hacer que tres terminales funcionen como entrada de datos, dos como salidas de control y otro grupo de cinco terminales como entrada de datos. Cada uno de estos grupos de terminales deberá estar asociado a una señal de habilitación de las seis de las que disponemos, de esta manera el terminal de entrada y salida podrá funcionar de manera bidireccional dependiendo de la señal que seleccione el multiplexor.

También habrá dos señales de realimentación que irán hacia la PIA, o al registro de entrada de la macrocelda, este registro hace que los cálculos se han mas veloces, ya que nos ahorramos pasar la señal por la PIA

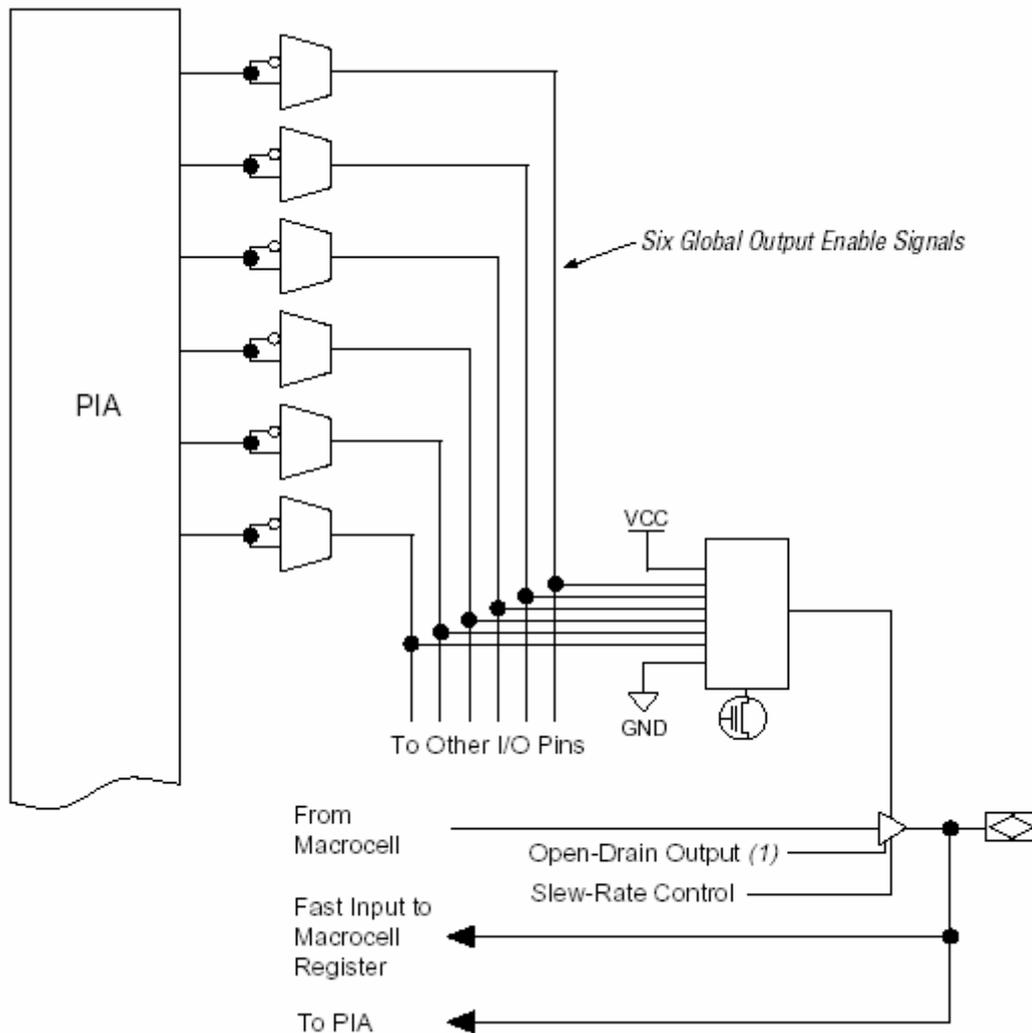


Figura 10: Diagrama de funcionamiento del bloque de entrada/salida

### 3.2.3 Matriz de interconexión programable (PIA)

La matriz de interconexión programable es idéntica a todos los CPLD con arquitectura MAX, conectando los bloques lógicos a los bloques de entrada/salida, habilitando estas entradas/salidas para que puedan ser utilizados por los diferentes LABs que conformen el CPLD.

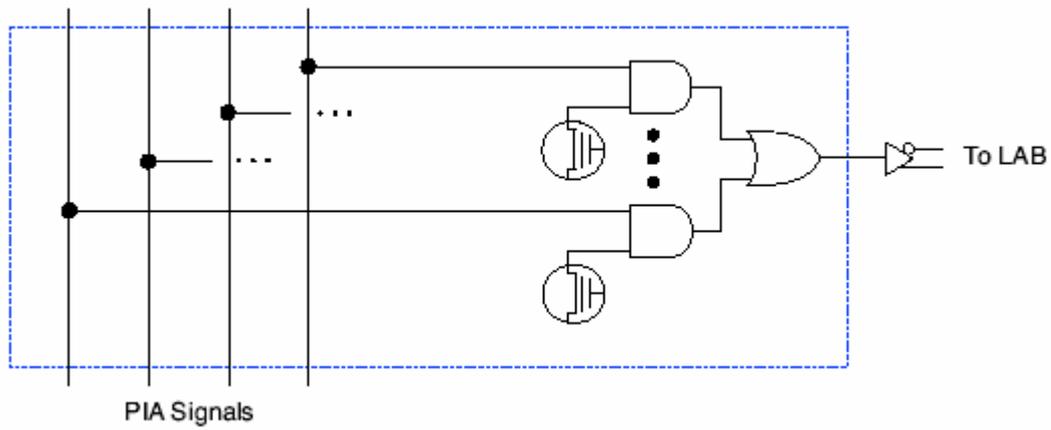


Figura 11: Diagrama de funcionamiento de la PIA

En este gráfico podemos ver como la señal de la PIA se distribuye hacia el LAB, de tal manera, que una célula EEPROM controla una de las dos entradas de la puerta AND, las cuales hacen la selección de la señal de la PIA para poder conducirla al bloque lógico que corresponda.

## **4 DISEÑO Y COMPONENTES**

### **4.1 Diseño y componentes que confeccionan el entrenador**

El entrenador está formado por un conjunto de entradas y salidas de información que se han establecido para poder imitar a la realidad poniendo diferentes elementos que en ella se pueden encontrar, como pulsadores, microrruptores o leds que a la hora de estudiar el programa a ejecutar en el CPLD, irá muy bien para obtener la respuesta del programa a dichas entradas y salidas.

Con esta respuesta se consigue saber si el programa que hemos implementado en el entrenador es viable o no, y si debemos introducir cambios en el programa.

El entrenador se ha configurado de manera modular, es decir, cada elemento de la PCB es independiente, sólo comunicados entre ellos por la tensión de alimentación y la masa. Esto se ha hecho para poder utilizar los módulos adecuados a cada proyecto, sin necesidad de que haya de estar toda la placa conectada.

Para ello se han creado conectores que unen los módulos para poder determinar la respuesta de los programas que se prueben en él.

También se ha dispuesto el método de programación en sistema, para poder programar el dispositivo, sin tener que extraerlo de su zócalo en la PCB, mediante el protocolo denominado JTAG.

El entrenador dispone de un CPLD de altera de 84 pins que será el encargado de interpretar los programas que se quieran probar.

Además se dispone de 3 relojes de diferente frecuencia para poder seleccionar la velocidad del CPLD sin necesidad de tener que implementar divisores de frecuencia internos.

Estas frecuencias se podrán seleccionar mediante *jumpers*.

Para poder interactuar con el entrenador, éste debe incorporar diversos componentes de entrada para poder facilitar el trabajo con el CPLD.

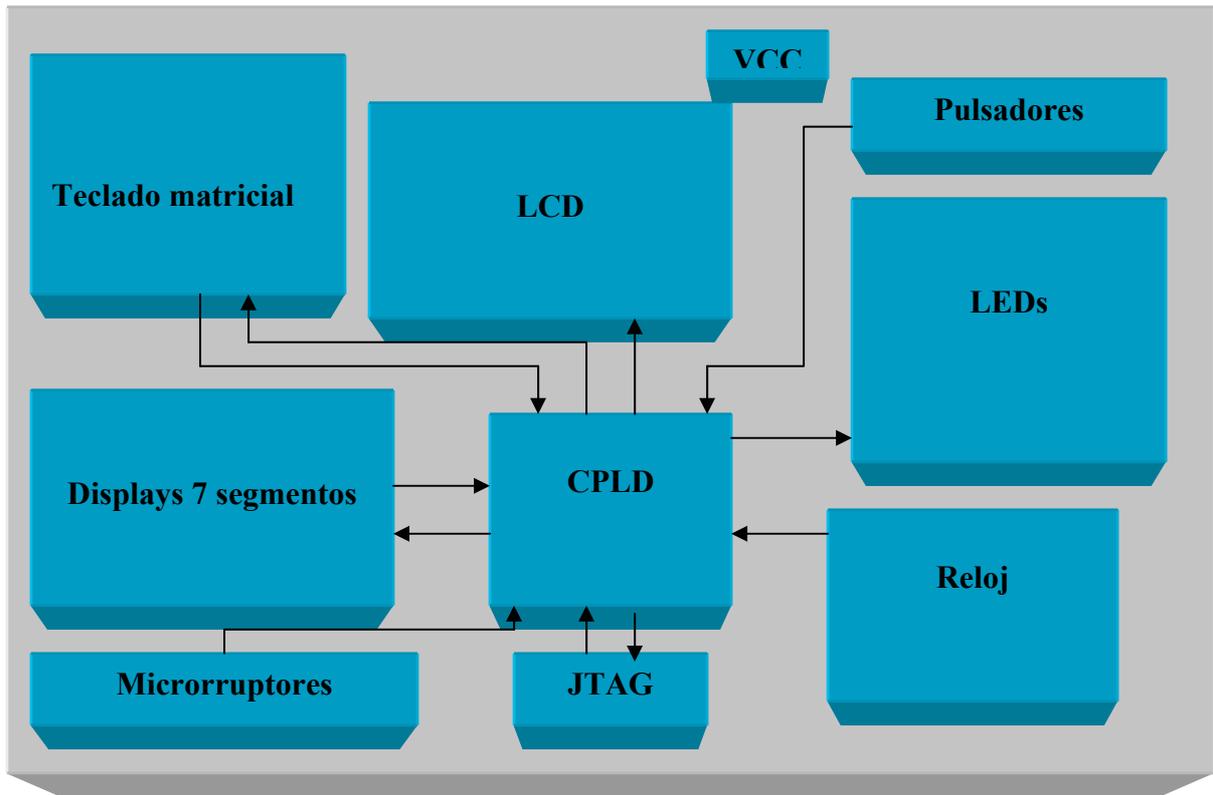
Las entradas de datos del entrenador incluyen:

- Un teclado matricial
- Pulsadores
- Interruptores del tipo DIL.

Para poder ver si los proyectos que se evalúan en el entrenador funcionan, tenemos diversas salidas que nos proporcionaran la información adecuada para verificar el correcto funcionamiento de estos.

Las salidas del entrenador están formadas por:

- 4 *displays* de 7 segmentos
- 1 pantalla LCD
- Varios leds



**Figura 12:** Esquema de los diferentes módulos que conforman la tarjeta de evaluación

En este diagrama se puede ver como están dispuestos los diferentes elementos que conforman la tarjeta de evaluación del CPLD. Y también como interactúan con éste.

## 4.2 Características del CPLD

EL CPLD utilizado en el entrenador es el EPM7128SLC84-15.

Este es el significado de la etiqueta que tiene el CPLD

EPM.- Indica a la familia que pertenece en este caso la familia MAX

7.- Este dispositivo pertenece a la familia 7000

128.- Es el número de macroceldas

SL.- Es el tipo de encapsulado

C.- Es la temperatura de trabajo en este caso es la temperatura comercial entre 0-70°

84.- Es el número de pins que dispone

-15.- Es la velocidad de respuesta en nanosegundos

Este CPLD forma parte de la familia MAX (*Matrix Array Matrix*), exactamente de la segunda generación de los Max7000, los denominados Max7000S.

El CPLD tiene 128 macroceldas agrupadas en bloques de 16, es decir, un bloque lógico o LAB según la denominación de Altera, contiene 16 macroceldas, por lo tanto tendremos que el CPLD utilizado tiene 8 bloques lógicos que estarán interconectados a los bloques de entrada y salida mediante la PIA.

El número de pins de entrada/salida que se usará será de 64, debido a que el resto de pins están dedicados a otras funciones.

Algunas características que posee esta familia y en especial nuestro CPLD son:

- Tiene hasta 2500 puertas.
- Está fabricado con tecnología MOS, idénticos a los utilizados por las memorias pasivas EEPROM (*Electrically-Erasable Programmable Read-Only Memory*)
- El tiempo de propagación entre terminales de entrada/salida es de 5ns lo que permite alcanzar frecuencias de 175.4Mhz.

- Es capaz de configurarse en sistema (ISP) gracias a la utilización del estándar JTAG.
- Es suministrado dependiendo del patillaje en diferentes encapsulados, en este caso se ha utilizado el tipo PLCC, con terminales tipo J.
- Es capaz de controlar la pendiente de cambio de los terminales de salida (*Slew Rate*). Con una baja velocidad podemos reducir el ruido que se podría introducir en el sistema, como contrapartida habrá un retardo de 5ns. El CPLD utilizado dispone de la característica de poder controlar el *slew rate* pin a pin.
- Posee 2 relojes globales.
- 1 bit de seguridad para que el programa no pueda ser leído, solo ejecutado para así poder proteger la propiedad intelectual
- Modo de consumo programable que permite reducir el consumo en un 50% de cada macrocelda. Esto se consigue debido a que algunas aplicaciones sólo necesitan una fracción de todas las puertas para operar a la máxima frecuencia, irá más lento y consumirá menos, la familia 7000 permite la programación de la velocidad de las macroceldas para configurarlas en alta velocidad o en bajo consumo, si las configuramos en bajo consumo todos los tiempos de respuesta aumentarán

Estas son algunas de las características de esta familia, hay muchas más, y para entrar en detalle se pueden encontrar en el *Data Sheet* del CPLD que proporciona ALTERA.

### **4.3 Tabla resumen de las conexiones de entrada/salida del CPLD**

La flexibilidad que proporciona el utilizar cables no rígidos para conectar los diferentes módulos que configuran el entrenador con el CPLD, permite no tener pins de éste dedicados en exclusiva a una sola utilidad, pudiendo elegir qué elementos de la placa se quieren conectar en función del programa que se quiera evaluar.

Sólo se debe tener en cuenta que el CPLD por su propia naturaleza tiene unos terminales dedicados a la alimentación y otros dedicados a su programación, por ello estos no se podrán utilizar como entradas o salidas.

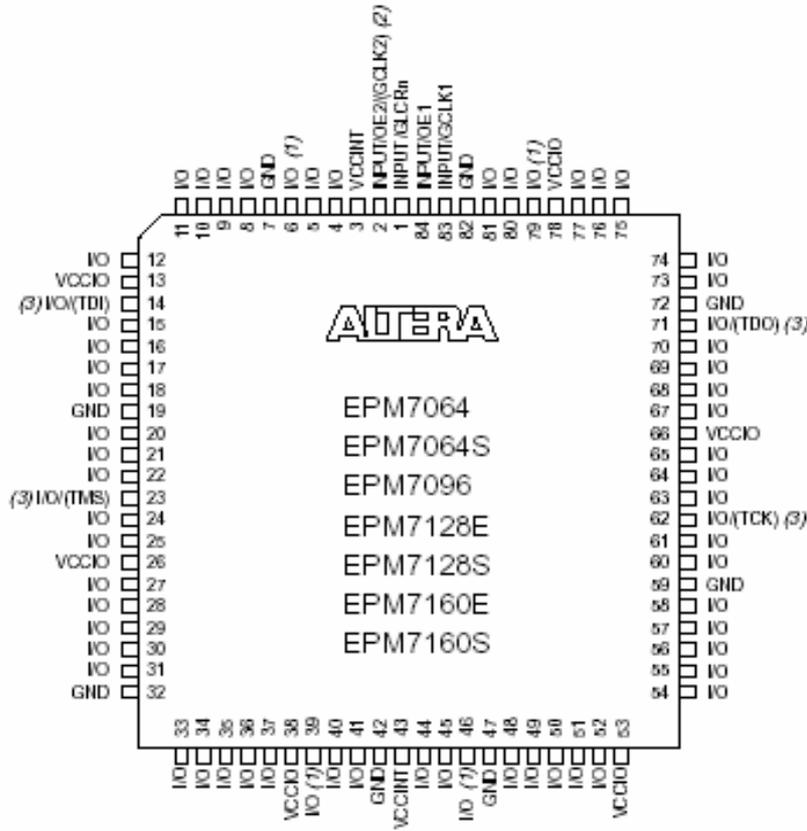


Figura 13: Encapsulado utilizado por la tarjeta de evaluación del CPLD, y numeración de los terminales

En las siguientes tablas se procede a numerar que pins del CPLD se usan en exclusiva y para qué:

<b>Pin</b>	<b>Utilidad</b>
14	TDI
23	TMS
62	TCK
71	TDO

**Tabla 1:** Numeración de los pins de programación

Los pins 14, 23, 62, 71 se utilizan para la programación en sistema del CPLD.

<b>Pin</b>	<b>Utilidad</b>
83	GCLK1
1	GCLRn

**Tabla 2:** Numeración de los pins del reloj

El pin 83 y 1 forma parte del reloj, siendo la señal de entrada del *clock* general y el *clear* respectivamente.

<b>Pin</b>	<b>Utilidad</b>
84	OE1
2	OE2

**Tabla 3:** Numeración de los pins de habilitación

Los pins 84 y 2, habilitan las líneas destinadas a las entradas del CPLD

<b>Pin</b>	<b>Utilidad</b>
42, 82	GNDINT
7, 19, 32, 47, 59, 72	GNDIO
3, 43	VCCINT
13, 26, 38, 53, 66, 78	VCCIO

**Tabla 4:** Numeración de los pins de alimentación

Y el resto son pins dedicados a la entrada de tensión y a la masa.

Sobran 64 pins que estarán dedicados a las entradas/salidas de la información, estos serán los utilizados para conectar las diferentes partes que conforman el entrenador con el CPLD.

## **4.4 El Estándar JTAG**

Hay otros métodos de programar un CPLD, pero al ser este tipo de programación la utilizada, y por sus múltiples beneficios la más estandarizada por los CPLD que hay en el mercado, la atención se centrará sólo en este método.

Para poder grabar el CPLD con el programa que se ha diseñado debe usarse en el entrenador el cable de conexión *BYTEBLASTER* que proporciona Altera para poder programarlo.

Un CPLD de la familia MAX7000S se puede programar de dos maneras:

- Fuera de sistema
- En sistema

En el caso de estudio se prepara el entrenador para que el CPLD sea programado en sistema, de esta manera será posible programar, borrar, reprogramar y comprobar el funcionamiento directamente en el entrenador sin necesidad de quitar el CPLD del zócalo. Esto facilita el diseño y la disminución del tiempo a la hora de hacer las pruebas.

El estándar JTAG se ajusta a la norma denominada IEEE-1149.1

### **4.4.1 EL STD.IEEE-1149.1**

En 1980 un grupo de compañías europeas se unió con el fin de estandarizar el sistema de comprobación de los dispositivos lógicos programables en sistema, al cabo de un año compañías de Asia y América se unieron para apoyar la creación del estándar dando como resultado la norma IEEE-1149.1 y en 1993 se perfeccionó la norma dando como resultado la nueva norma IEEE-1149.1-1993

## **4.5 Estructura del bus JTAG**

Cuatro son los terminales básicos que debe tener el bus de datos. Se identifican como:

- TDI .- Entrada de datos (*Test Data Input*)
- TDO .- Salida de datos de prueba (*Test Data Output*)
- TMS .- Selección del modo test (*Test Mode Select*)
- TCK .- Reloj (*Test Clock Input*)

El caso de estudio utiliza el bus de conexión *BYTEBLASTER* de Altera, será necesario disponer de 2 terminales de más, dedicados a la entrada de tensión y a la masa. Como indican las especificaciones, la tensión de alimentación será de 5V y esta tensión la suministrará la PCB.

Los modelos que se podrán programar en el entrenador aparte del ya mencionado serán todos los CPLD de la familia MAX7000S, FLEX10K, FLEX 8000, FLASHlogic, MAX9000 y todos los PLDs que dispongan de la opción ISP

Este bus se conectará al puerto paralelo (de impresoras) del PC.

### **4.5.1 Entrada de datos**

Este terminal es una entrada de datos series para las instrucciones, la programación del dispositivo y la comprobación de funcionamiento. La transmisión se efectuará cuando haya un flanco de subida proveniente del reloj del sistema.

### **4.5.2 Salida de datos de prueba**

Es una salida de datos para la programación del sistema, la comprobación del funcionamiento y las instrucciones, esta transmisión de datos se efectuará cuando haya un flanco de bajada del reloj del sistema, por lo tanto este valor estará a alta impedancia cambiando cuando el valor se modifica.

## 4.6 Salidas del entrenador

### 4.6.1 *Displays* de 7 segmentos

Los *displays* de 7 segmentos están formados por 7 leds. Cada led representa un segmento, estos se deben montar de tal manera que cada segmento se ilumine por separado pudiendo combinarlos para crear números, del 0 al 9 o del 0 a la F en el caso de valores hexadecimales.

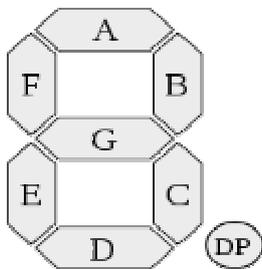


Figura 14: *Esquema de los segmentos que conforman un Led 7 segmentos*

Cada segmento del *display* tiene un valor que identifica su posición. Al combinar estos valores se pueden crear las combinaciones que den como resultado el número deseado.

Hay dos posibles maneras de conectar los *displays*, en modo ánodo común o en modo cátodo común.

En modo ánodo común hay que poner todos los ánodos de los leds conectados a la fuente de alimentación, mientras que los cátodos estarán unidos por una resistencia a masa para poder limitar el paso de corriente por los diferentes leds.

Cuando se conecta en modo cátodo común, el cátodo es el que está siempre conectado a la tensión, mientras que los ánodos están conectados a masa a través de resistencias que limitan la corriente.

### 4.6.2 Conjunto de resistencias

El conjunto de resistencias usado entre el *display* de 7 segmentos y las salidas del conjunto de la visualización dinámica, está enfocado a limitar el paso de corriente por los leds que componen cada uno de los segmentos. Para conocer el valor óhmico de las resistencias sólo hay que aplicar la Ley de Ohm.

Se debe asegurar la integridad del *display* teniendo en cuenta que la intensidad que pase por cada uno debe ser lo suficientemente elevada para poder visualizar-lo, teniendo en cuenta que estará poco tiempo encendido, pero asegurándonos que no pasa tanta intensidad como para poder poner en riesgo alguno de los segmentos que conforman el *display*.

La intensidad que ha de pasar por cada segmento, suele variar dependiendo del fabricante pero se puede suponer una intensidad de entre 10 y 20 mA, por lo tanto el valor óhmico será de:

$$V_i / I = 5V / 10mA = 500\Omega$$

Este valor no está normalizado por lo tanto no se comercializa, así que cogeremos un valor de 470  $\Omega$  (amarillo – violeta – negro).

### 4.6.3 Transistores PNP

Estos transistores se utilizarán para hacer el control del sistema de visualización dinámica, en nuestro caso cada vez que se aplica un 0 lógico en la base del transistor, este deja pasar tensión del emisor al colector provocando que los segmentos de los *displays* se activen y desactiven rápidamente, dependiendo de cuál de los cuatro transistores tenga en su base un 0 lógico.

### 4.6.4 Leds

El uso de señales luminosas, se ha pensado para poder interpretar las señales de salida de los proyectos que se ejecutan en el entrenador, por este motivo y para este fin se ha implementado el uso de Leds

### **4.5.3 Selección del modo test**

Es una entrada para seleccionar el modo de funcionamiento que se va a realizar con el dispositivo, su valor determina la secuencia de trabajo del dispositivo, si es 0 corresponderá al modo de verificación, si es 1 al funcionamiento normal.

### **4.5.4 Reloj**

Es la señal de la cual el resto del sistema se sincronizará para transmitir los datos y así poder programar el CPLD.

Un led es un semiconductor que emite luz cuando se polariza directamente y pasa por él una corriente. Dependiendo del material fabricado, éstos emiten un espectro de luz diferente.

Para que funcione un led debe pasar una corriente determinada en un sentido determinado, ánodo – cátodo. Y para evitar que éste se quemara, debe limitarse el paso de corriente a través de él, mediante la conexión de una resistencia entre la salida del cátodo y el pin de salida.

Para limitar la corriente se debe usar la Ley de Ohm, de la siguiente manera:

$$V_i - V_{led} / I_{led} = 5v - 2v / 10mA = 300\Omega$$

Se usa por lo tanto una resistencia de 300  $\Omega$  (naranja-negro-marrón) para limitar la corriente.

Se ha escogido una tensión de led de 2V que es lo que normalmente suelen usar los leds.

#### 4.6.5 Pantalla LCD

El uso de una pantalla LCD, sirve para poder mostrar menús, datos de diversa índole o mensajes que pueden usarse a la hora de interpretar los resultados de la programación del CPLD.

Utilizando esta pantalla nos podemos ahorrar el uso de transistores, resistencias, descodificadores, *display* de 7 segmentos, etc... Que hacen más costoso y más complicado el entrenador.

El *display* utilizado en este caso es un *display* de 4 filas y 16 caracteres por fila. Funciona a una tensión máxima de 7 V y tiene entradas de 4/8 bits en paralelo para poder controlar el generador de 192 caracteres que lleva incorporado.

#### 4.6.6 Pins de control del *Display* LCD

El *display* LCD está formado por 16 pins que se conectan a la PCB de la siguiente manera:

Pin1.- Vss	Este pin está dedicado a masa.
Pin2.- Vdd/Vcc	Entrada dedicada a la alimentación del <i>display</i> .
Pin3.- Vo	Ajuste de la iluminación de contraste.
Pin4.- Rs	Dice si el valor es un comando (RS = 0) o un carácter (Rs = 1).
Pin5.- R/W	Dice si estamos enviando datos (R/W = 0) o está leyendo el estado del <i>display</i> (R/W = 1).
Pin6.- E	<i>Enable</i> selecciona el <i>display</i> a utilizar.
Pin7-14.-	Bus de datos siendo el bit DB0 el de menos peso y el DB7 el de mayor peso.
Pin15.- A	<i>Backlight</i> alimentación de los leds del LCD (ánodo).
Pin16.- K	Masa (cátodo).

En el *display* LCD del entrenador se le han incorporado dos resistencias variables.

La primera es de  $10\text{K}\Omega$ , y se ha utilizado para regular el contraste de la pantalla, para darles más o menos luz. Esta resistencia variable se conecta entre la alimentación y la masa y el regulador de la resistencia es la que va al pin nº 3 o Vo.

El tercero es de  $100\text{K}\Omega$  y se coloca entre la salida del pin nº 16 y masa controlando así la intensidad que pasa por los leds que conforman la retro-iluminación.

## 4.7 Entradas del entrenador

### 4.7.1 Teclado matricial

Se ha implementado un teclado matricial, para la introducción de valores decimales. En concreto se pueden introducir 12 valores, 9 valores decimales y 2 valores alfanuméricos (\* y #).

Se ha escogido un teclado matricial porque con pocas señales se pueden controlar varias teclas, en concreto con “n” señales se pueden controlar “2 ^ n” teclas.

El funcionamiento de un teclado matricial es el siguiente, a la hora de pulsar una tecla, una fila y una columna se ponen en contacto con lo que se deduce el valor de la tecla oprimida.

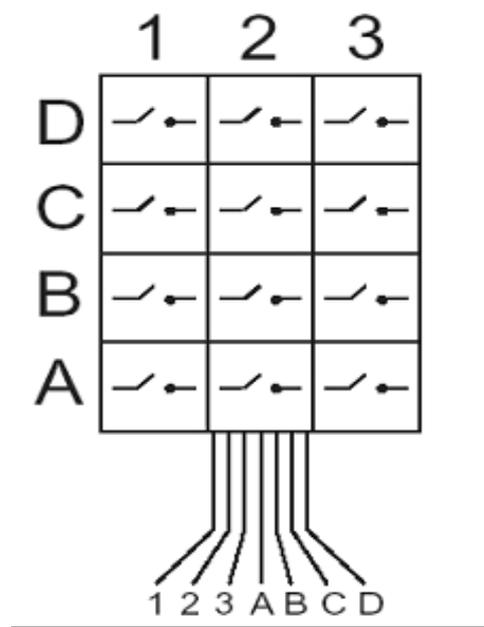


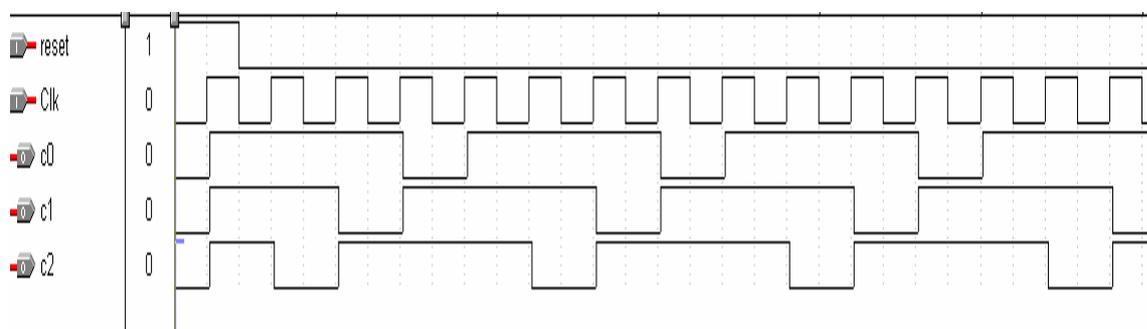
Figura 15: *Diagrama de funcionamiento de un teclado matricial*

Se conectan las columnas a las entradas del CPLD y las filas se conectan a un *pack* de 4 resistencias de  $1k\Omega$ , conectadas en posición *pull-up*, con esta conexión las filas están siempre a un valor lógico alto (1).

Se conoce qué fila se ha pulsado cuando ésta está a valor bajo o “0”, por lo tanto con las columnas pasa lo mismo, cuando una columna está a valor lógico bajo es porque ha sido activada. Entonces se puede deducir el valor de la tecla pulsada.

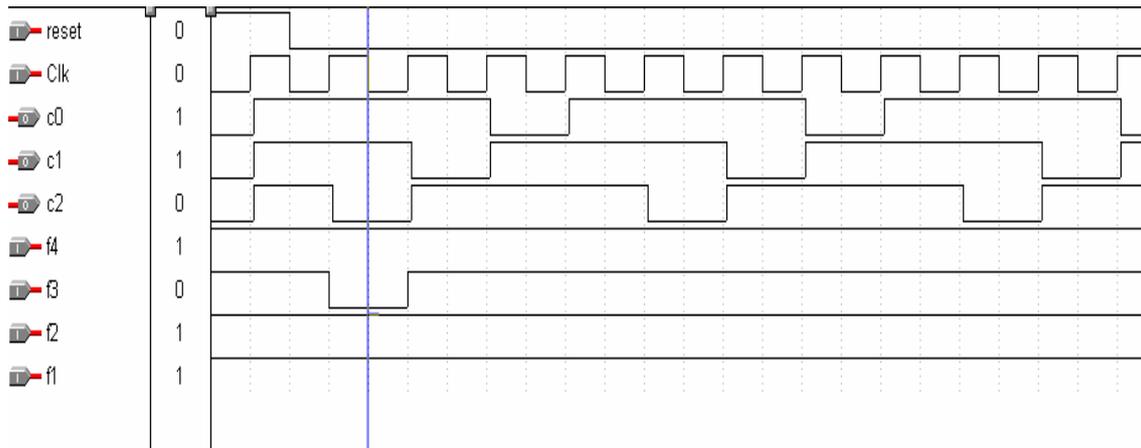
Por lo tanto para el funcionamiento del teclado se ha de crear una señal que explora las columnas del teclado de manera que cada cierto tiempo una de estas columnas está a nivel bajo, es decir que está activa.

El proceso de exploración tiene lugar cada vez que el entrenador está conectado a la red, se inicia un proceso de envío de flancos de alta impedancia para cada una de las columnas, como muestra la siguiente figura, para determinar qué columna está activa.



**Figura 16 Señal de barrido de columnas**

Para aclarar el funcionamiento del programa aquí se muestra un ejemplo:



**Figura 17:** *Ejemplo de la pulsación de una tecla*

El proceso se inicia cuando se detecta un nivel diferente a 1111 en alguna de las filas.

Los estados posibles son:

- Fil. [3...0].- 1110 la tecla pulsada pertenece a la primera fila.
- Fil. [3...0].- 1101 la tecla pulsada pertenece a la segunda fila.
- Fil. [3...0].- 1011 la tecla pulsada pertenece a la tercera fila.
- Fil. [3...0].- 0111 la tecla pulsada pertenece a la cuarta fila.

Una vez se ha determinado la fila y la columna que se está usando, se pasa la información a un decodificador que nos indica que tecla ha sido pulsada.

En este ejemplo se ha pulsado la tecla que tiene como código 1101011, que después de descodificar el código, obtenemos que pertenece a la tecla del nº 6.

### 4.7.2 Pulsadores

La utilización de pulsadores en el entrenador da la posibilidad de utilizarlos como entradas de señales, que permiten interpretar el encendido de una máquina, la pulsación de una tecla o simplemente pulsaciones de un contador. Las posibilidades son varias dependiendo de las pruebas que se quieran ejercer en el entrenador.

Estos pulsadores pueden dar valores de entre 0 y 5V dependiendo de si éste está pulsado o no, por lo tanto da la posibilidad de enviar datos al CPLD sólo con la alimentación de la placa.

Se han implementado para este hecho 20 pulsadores táctiles con el fin de poder usarlos como mejor se necesiten en cada proyecto.

### 4.7.3 Interruptores tipo DIL o Microrruptores

Se han utilizado 2 interruptores tipo DIL de 8 vías, para poder representar hasta 256 ( $2^n$ ) valores por cada interruptor de estas características. Es decir que conjuntamente se pueden representar hasta 512 combinaciones utilizando como señal la alimentación de la PCB.

Se han configurado mediante un *pack* de 8 resistencias de  $1K\Omega$  en posición de *pull-down* para asegurar que fija un 0 lógico y no tiene valores indefinidos cuando el interruptor está a medio camino entre el 1 y el 0.

#### **4.7.4 Conectividad de los diferentes módulos que conforman el entrenador**

El entrenador es frecuentemente utilizado para su uso en distintas aplicaciones, indistintamente de las entradas y salidas que éste tenga preparadas para su uso.

A veces no es necesario usar el *display* de siete segmentos pero sí el *display* LCD, o necesitamos algunos pulsadores pero no los microrruptores.

Por ello se ha optado por un uso flexible de los componentes que configuran el entrenador, siendo el CPLD el centro de todas las conexiones ya que es el encargado de hacer funcionar todo el sistema.

Para esto se ha optado por conectar los distintos módulos que configuran la PCB al CPLD mediante cables con conectores hembra.

Haciendo esto es posible elegir qué módulos son necesarios para utilizar en el proyecto que se está evaluando y cuáles son prescindibles, ahorrando así el trabajo de tener que conectarlos y permitiendo elegir al usuario multitud de combinaciones. También de esta manera se pueden hacer pruebas tales como poder cambiar la configuración inicial del proyecto que se está evaluando para poder evaluarlo mejor o perfeccionándolo con diferentes módulos que antes no se habían agregado.

#### **4.7.5 Conexión de los diferentes módulos**

Los cables utilizados para la conexión de los distintos módulos están formados por cables flexibles de 20 cm de longitud, con terminales hembra de engaste para la conexión.

Cada módulo tiene en sus entradas y salidas conectores macho.

La forma de conexionado de los diferentes módulos está basada en una conexión pin a pin teniendo como elemento común el CPLD.

#### 4.7.6 Disposición de los relojes

Los relojes de la placa se han dispuestos para enviar la señal de *Clock* a los diferentes dispositivos que la requieran, Estos relojes son 2 circuitos 555 en modo astable, y un oscilador de cuarzo. Es interesante tener distintas señales de *Clock* en el entrenador para poder seleccionar la más adecuada a las necesidades del programa que se quiere probar, sin la necesidad de tener que implementar en el CPLD un divisor de frecuencia, aunque si fuese necesario se puede realizar.

#### 4.7.7 Oscilador de cuarzo

Éste está formado en su núcleo por un cristal de cuarzo, que es el componente que hace el control de frecuencia de los osciladores de este tipo, convirtiendo las vibraciones mecánicas en tensión a una frecuencia específica.

Esto ocurre debido al efecto piezo-eléctrico, se denomina así a la particularidad de crear electricidad a partir de una presión mecánica.

Por las propiedades del cuarzo, es el material más usado para crear estos cristales, ya que dan una respuesta más controlada.

En este proyecto se ha utilizado un cristal de cuarzo de 4 pins. Y una frecuencia de 1 Mhz.

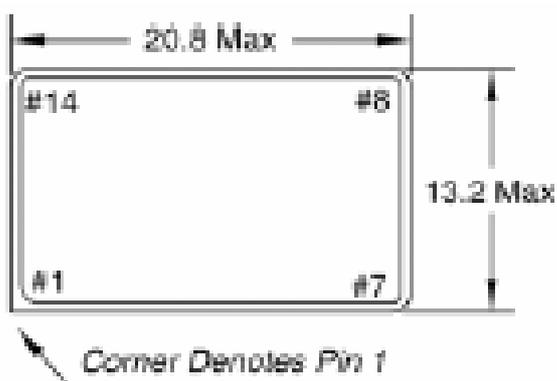


Figura 18: Diagrama de conexión de un cristal de cuarzo de 4 terminales

### 4.7.8 Oscilador 555

Se ha utilizado este circuito integrado porque es muy estable y temporiza con gran precisión, además se puede configurar para que funcione como un oscilador.

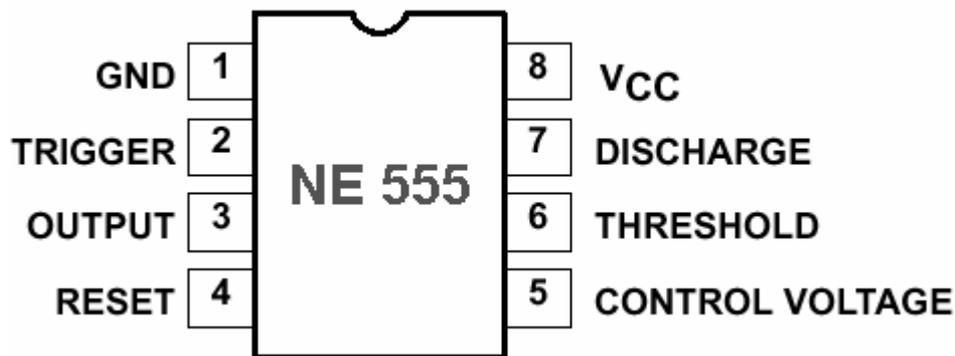


Figura 19: Descripción de los terminales del 555

En este diagrama se puede ver que el componente dispone de los típicos terminales de control, como vcc, masa, salida y reset. También dispone de la salida *threshold*, que permite finalizar la temporización cuando se activa; o el terminal de *trigger*, que cuando recibe un impulso negativo hace que el condensador externo del pin número 7 se descargue, este condensador permite controlar el tiempo de la temporización.

En el entrenador se han dispuesto 2 circuitos 555 en modo astable, que puede proporcionar una frecuencia de 20 a 20Khz, mediante una onda periódica cuadrada. Para disponer de varias frecuencias se ha sustituido la resistencia número 2, por una resistencia variable y así poder variar la frecuencia que nos aportan los 555 en la salida

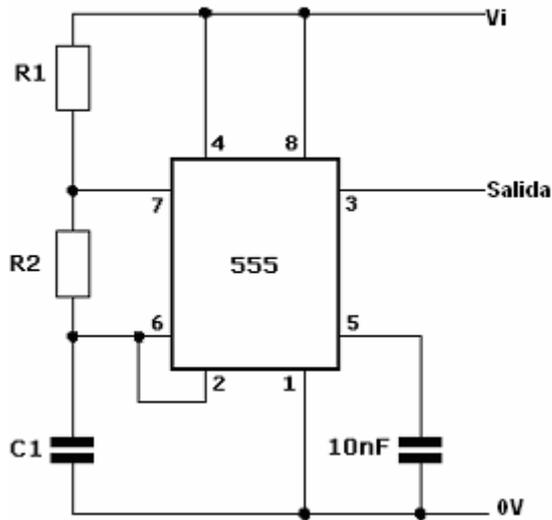


Figura 20: Esquema de conexión del 555

Que el micro 555 esté funcionando en modo astable, indica que la tensión de salida varia a la frecuencia indicada, variando continuamente entre dos valores permaneciendo un tiempo T1 en el primero y un tiempo T2 en el segundo cuyo periodo será de  $T = T1+T2$ .

Por lo tanto variando T1 y T2 se consigue la frecuencia deseada.

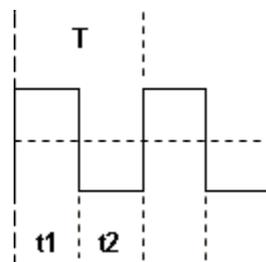


Figura 21: En este diagrama observamos los tiempos a los que hacemos referencia

El tiempo en que la salida está a nivel alto viene dado por la siguiente fórmula:

$$T1 = 0.693 (Ra +Rb)C$$

El tiempo en que la salida está a nivel bajo viene dado por la fórmula:

$$T2 = 0.693 Rb C$$

El tiempo Total o periodo viene dado por la suma de ambos tiempos.

$$T = T1 + T2 = 0,693 (Ra + 2Rb) C$$

Y por último la frecuencia de oscilación es:

$$F = 1 / T = 1.44 / (Ra + 2 Rb) C$$

También existe la posibilidad de hacer funcionar el 555 en modo estable. No es útil en este caso porque sólo se genera un único pulso de salida.



## **5 MANUAL DE CREACIÓN Y FUNCIONAMIENTO**

### **5.1 Pasos a seguir para crear la PCB**

A continuación tiene lugar la creación de la PCB. Una vez ya se ha creado el diseño previo necesario de las pistas, utilizando *OrCad Layout*.

Para empezar se necesita una placa PCB fotosensible positiva. Se corta usando una guillotina a la medida deseada.

Se debe imprimir en transparencia plástica el fichero con los circuitos generados gracias al programa de diseño, esto se emplea sólo con prototipos dada la baja resolución que se consigue con la impresión.

Para que haya más contraste entre el blanco y el negro se puede hacer una fotocopia en papel transparente de la impresión que se ha sacado anteriormente.

Existen tres procedimientos para crear una PCB dependiendo de la manera escogida a la hora de transferir el diseño a la placa. Este procedimiento se basa en dibujar en el cobre, las pistas o *pads*, ya sea manual o automáticamente:

- **Manual:** se dibujan las pistas usando un rotulador negro o plantillas de símbolos transferibles.
- **Fotográfico:** Se compra una placa presensibilizada que trae una capa totalmente uniforme y sin imperfecciones preparada para la transferencia.
- **Serigráfico:** mediante maquinaria.

El mejor método para hacer prototipos es el fotográfico.

Se coge la impresión o la fotocopia en transparencia que se ha realizado previamente de las pistas y se coloca mediante celo transparente encima de la placa presensibilizada, para evitar que ésta se desplace.

Una vez realizada esta tarea, llega la etapa de insolado de la placa, cuyo objetivo es transferir las pistas de la transparencia a la placa, debilitando así las zonas en las que se quiere eliminar el cobre.

Para este paso es necesario disponer de una insoladora y es importante evitar que haya bolsas de aire entre la transparencia y la placa.

Una vez terminado el proceso de insolación se inicia el revelado de la PCB. Es un proceso que deja al descubierto el cobre en los sitios donde penetra la luz, eliminando la película fotosensible que hay encima.

Para este proceso se debe introducir la placa en un líquido revelador basado en el hidróxido sódico, NaOH+agua. (Esta solución se encuentra en tiendas de electrónica).

Se tiene que controlar el proceso para que la eliminación de la capa fotosensible sea uniforme, además hay que vigilar que la solución no sea muy elevada y que la placa no esté mucho tiempo debajo de ésta, ya que el revelador puede atacar también las partes que no han sido sensibilizadas provocando que la placa se pierda.

Al cabo de un rato se procede a la extracción de la placa y se lava con abundante agua.

Llegados a este punto se puede ver plasmado en la PCB el dibujo de las pistas. Una vez hecho esto, se pasa al llamado atacado químico, usado para la eliminación de las zonas de cobre no protegidas de la placa.

La solución utilizada es una solución ácida que puede ser de dos tipos dependiendo de la composición química.

- **Atacador lento:** Compuesto de Cloruro férrico ( $CL_3Fe$ ). Este proceso es más rápido si se calienta la solución
- **Atacador rápido:** Compuesto por 1/3 de Ácido clorhídrico o sulfúrico + 1/3 de agua oxigenada + 1/3 de agua

Estos productos químicos también se pueden conseguir en una tienda de electrónica.

Lo siguiente es controlar la tapa del atacado de igual manera que en la etapa del revelado, ya que sino se puede llegar a perder la placa.

Una vez haya desaparecido el cobre de la placa, ésta debería quedar en un tono verdoso y las pistas de color dorado mientras el resto del cobre debería tomar un color rosa oscuro.

Seguidamente se aclara bien la placa con abundante agua, y si quedan restos del producto químico utilizado, hay que introducirlo en una solución de agua + amoniaco a partes iguales. Y para la limpieza final pasar un paño humedecido con alcohol con el fin de eliminar por completo la película fotosensible que queda sobre las pistas.

Una vez obtenida la PCB se procede a mecanizarla mediante el uso de un taladro, hay que asegurarse que el diámetro de los agujeros es asequible a la hora de insertar los componentes. Una vez hechos los taladros se ponen los componentes y se pueden soldar obteniendo así el prototipo deseado.

## **5.2 Funcionamiento del entrenador**

Pasos a seguir para el funcionamiento del entrenador.

### **Primer paso:**

Pensar que módulos se necesitan para simular el programa.

### **Segundo paso:**

Ahora se deben interconectar los módulos necesarios para realizar las pruebas a las entradas y salidas del CPLD, mediante los cables de conexión descritos con anterioridad.

### **Tercer paso:**

Al programar el entrenador hay que configurar los pins de entrada/salida del CPLD, para así identificar que pins corresponden a las salidas y cuáles a las entradas de los diferentes módulos para poder casarlos con las conexiones que se han realizado en el paso anterior.

**Cuarto paso:**

Se debe conectar el cable de datos *BYTEBLASTER* al puerto paralelo del PC y de allí conectarlo al conector macho de 10 pins de la PCB, para así poder reprogramar el CPLD con el programa realizado y después probarlo en el entrenador.

**Quinto paso:**

Una vez conectado el cable *BYTEBLASTER* a la placa, es necesario conectar la alimentación del entrenador, mediante el conector de alimentación que se encuentra en la tarjeta de evaluación. Ésta se debe alimentar a una tensión de 5 V para hacer funcionar los módulos.

**Sexto paso:**

Interactuar con los módulos escogidos para realizar las pruebas en el entrenador.

**Séptimo paso:**

Ver las respuestas que se suceden en el entrenador una vez se ha procedido a la instalación del programa, para poder evaluar la respuesta del programa en el sistema que simula su viabilidad en la realidad.

**Octavo paso:**

Si las respuestas recibidas del sistema no son las correctas debe modificarse el programa original, y como consecuencia volver a instalar el programa modificado otra vez en el CPLD. Para ello se repetirán los pasos del cuarto al séptimo, hasta conseguir el resultado deseado.

## 6 PRÁCTICAS

### 6.1 Práctica 1: Visualización dinámica

#### 6.1.1 Objetivo

El objetivo de esta práctica es conocer el funcionamiento de los *displays* 7 segmentos, mediante la implementación de un sistema de visualización dinámica.

Este tipo de visualización se utiliza para poder visualizar cada dígito durante un instante de tiempo y a una velocidad tal, que gracias a la persistencia del ojo, el efecto final es que todos los dígitos están encendidos al mismo tiempo.

También sirve para reducir el espacio que ocuparía el control de la gestión del encendido de los cuatro *displays* en la placa, ya que con poca electrónica, utilizando este método se puede hacer funcionar los 4 *displays*. Por este hecho se ha decidido crear un sistema de visualización de control dinámico.

Se han implementado 4 *displays* de siete segmentos de tal manera que es posible visualizar números, bien por un cálculo hecho por el CPLD o por que se está utilizando como un reloj, etc...

#### 6.1.2 Enunciado

Se debe diseñar y simular el funcionamiento de un dispositivo capaz de interpretar el valor entrado mediante un microrruptor, en binario BCD, para que éste sea visualizado en los *displays* 7 segmentos en formato decimal.

#### 6.1.3 Hardware necesario

El Hardware necesario es:

- 4 *displays* de 7 segmentos
- 1 conjunto de 8 resistencias

- 4 transistores PNP
- 12 pins

## **6.2 Práctica 2: Contador programable**

### **6.2.1 Objetivo**

El objetivo de esta práctica es conocer el funcionamiento de las entradas y salidas del entrenador, haciendo funcionar varias de estas.

### **6.2.2 Enunciado**

Diseñar y hacer funcionar un contador, que cuando llegue al final de cuenta proporcione una señal de salida, con la que encender uno o varios leds, también deberá poder programarse dependiendo de cuántos pulsadores se hayan pulsado. Cada pulsador funcionará cuando haya un flanco de subida.

## **6.3 Práctica 3: Control del teclado matricial**

### **6.3.1 Objetivo**

El objetivo de esta práctica es que el alumno aprenda a utilizar un teclado matricial creando el sistema de control del mismo. Una vez hecho esto el alumno podrá enviar la información introducida por el teclado hacia el *display* LCD para así poder ver el valor que se ha entrado.

### **6.3.2 Enunciado**

Se debe realizar un programa para poder introducir valores numéricos utilizando el teclado matricial que está implementado en la tarjeta de evaluación.

### **6.3.3 Módulos necesarios para la creación del teclado matricial**

El sistema de control del teclado matricial, que se ha de implementar con el fin de reconocer la tecla pulsada en cada momento, debe estar formado por los siguientes módulos:

- Generador de señales - Genera la señal de exploración
- Contador - Hace las funciones de un contador
- Descodificador - Descodifica la señal de entrada
- Control - Controla el proceso

### **6.3.4 Generador de Señales**

El módulo generador de señales es el encargado de generar la señal de exploración del teclado, el cual activará una vez, cada periodo de tiempo, cada una de las columnas de las que está compuesto el teclado matricial.

Para crear este módulo se necesita una señal de *Clock*, encargada de dar la frecuencia con la que se genera la señal de barrido del teclado.

Como cada columna será activa a nivel bajo tenemos que la combinación:

- 011 - activa la primera columna
- 101 - activa la segunda columna
- 110 - activa la tercera columna

Por lo tanto se forma la siguiente tabla de valores que hace referencia al circuito de barrido.

Código de columnas			Respuesta	
C0	C1	C2	Q1	Q2
0	1	1	0	0
1	0	1	0	1
1	1	0	1	0

\*C0,C1,C2, hace referencia a las columnas.

Entonces:

$$C0 = \overline{Q0} \quad \overline{Q1}$$

$$C1 = \underline{Q0} \quad \underline{Q1}$$

$$C2 = Q0 \quad Q1$$

Que debe dar como resultado el siguiente esquema que genera la señal de barrido del teclado:

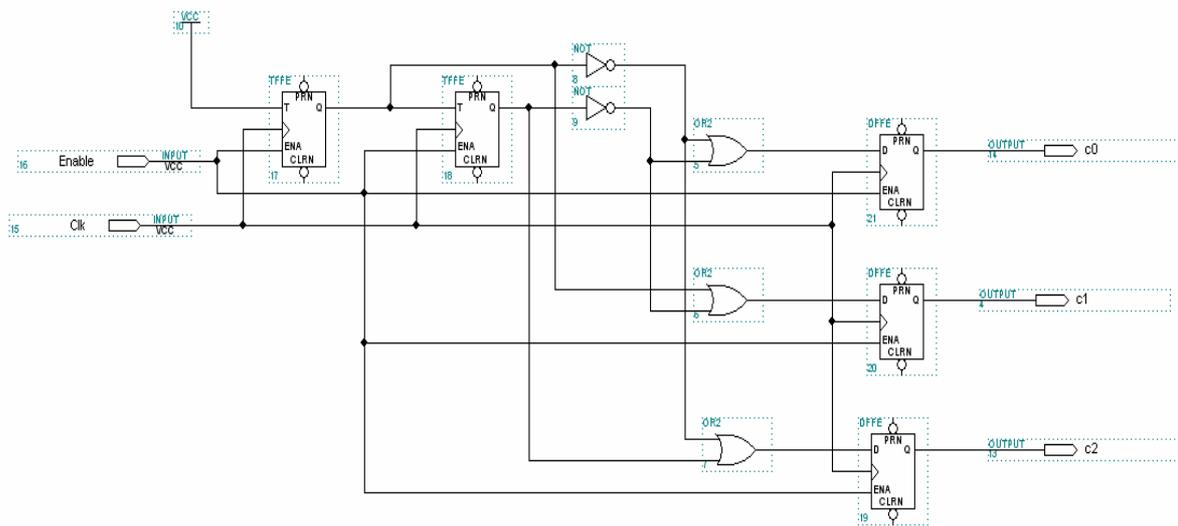


Figura 22: Ejemplo del generador de barrido

### 6.3.5 Contador

Se debe utilizar para detectar si la tecla pulsada en el teclado ha sido pulsada intencionadamente o por accidente. Esto es debido a que cuando se pulsa una tecla el contador se activa y empieza a temporizar, si éste ha acabado de temporizar y la tecla continúa pulsada entonces se da la señal necesaria para validar la pulsación de la tecla. De esta manera se evita que a la hora de pulsar la tecla haya rebotes.

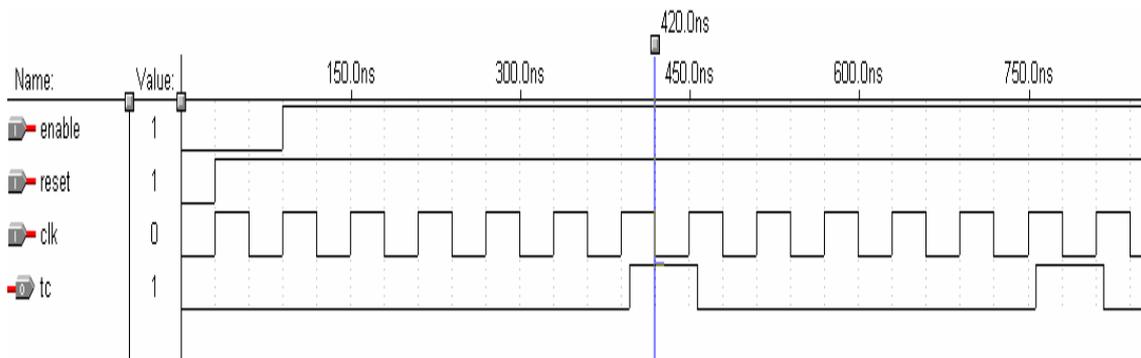


Figura 23: Ejemplo del funcionamiento de un contador

Una vez terminada la cuenta se activa la señal denominada TC; cuando esta señal está activa, y si la tecla sigue pulsada, se procede a validar la tecla.

Cambiando el valor de cuenta del contador se puede variar el tiempo que deberá mantenerse pulsada la tecla para que ésta sea validada.

### 6.3.6 Descodificador

El módulo denominado descodificador es el encargado de interpretar las señales que se han capturado provenientes de las columnas y las filas, y se encarga de transformarlas en un código interpretable por el *display* LCD o por el visualizador dinámico.

Para llevar a cabo esta interpretación se ha realizado la siguiente tabla.

Columnas			Filas				Valor
C1	C2	C3	F1	F2	F3	F4	
1	1	1	0	1	1	1	3
1	1	1	1	0	1	1	6
1	1	1	1	1	0	1	9
1	1	1	1	1	1	0	#
1	0	1	0	1	1	1	2
1	0	1	1	0	1	1	5
1	0	1	1	1	0	1	8
1	0	1	1	1	1	0	0
0	1	0	0	1	1	1	1
0	1	0	1	0	1	1	4
0	1	0	1	1	0	1	7
0	1	0	1	1	1	0	*

**Figura 24:** *Tabla de interpretación de valores del teclado.*

En esta tabla se puede ver perfectamente como funciona la señal de barrido, ya que las columnas son activas una vez cada cierto periodo de tiempo, mientras que las filas se activan en el momento que se pulse una tecla.

Una vez conocido el valor es posible codificarlo dependiendo de si se quiere en visualización dinámica, haciendo las combinaciones de segmentos necesarias para que aparezcan los números o *display* LCD.

### 6.3.7 Control

El módulo de control es el algoritmo encargado de gestionar todas las partes del sistema que conforman el funcionamiento del teclado matricial.

Este módulo no es más que una máquina de estados que funciona de la siguiente manera:

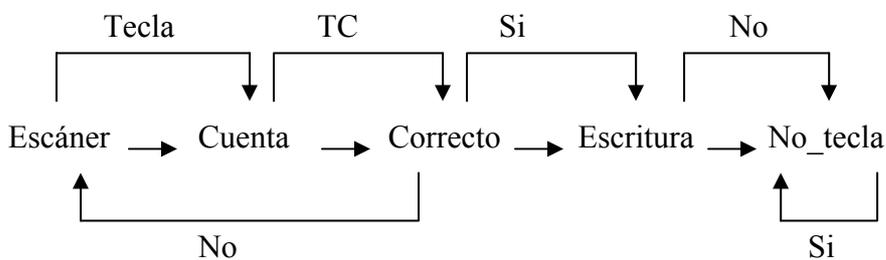


Figura 25: Esquema funcionamiento de la máquina de estados del teclado

- **Escáner:** Este estado es el encargado de empezar a explorar el teclado a la espera de que una tecla sea pulsada y así avanzar al siguiente paso. Por lo tanto si una tecla es pulsada la máquina avanzará al siguiente estado.
- **Cuenta:** Aquí se ha detectado el uso de una tecla por parte del usuario, la máquina activa el contador y al acabar la cuenta este envía una señal denominada TC.
- **Correcto:** La máquina se encarga de validar la tecla, si una vez finalizado el contador la tecla continua pulsada entonces se avanza hacia el estado de escritura.
- **Escritura:** La máquina de estado activa una señal de salida denominada WR, encargada de activar la salida del teclado hacia el *display* LCD o el *display* de 7 segmentos.
- **No\_tecla:** Hasta que la tecla no deja de ser pulsada no se vuelve al estado inicial.



## **7 MATERIALES DE FABRICACIÓN**

### **7.1 Listado de componentes que conforman el entrenador**

<b>Descripción del material</b>	<b>Cantidad</b>
<b>PCB</b>	
Placa fotosensibilizada para PCB (200x300)	1
<b>Resistencias</b>	
Integrados de 9 resistencias conectadas 1 K $\Omega$	3
Integrados de 9 resistencias conectadas 300 $\Omega$	3
Resistencia de 51 K $\Omega$	1
Resistencia de 5K1 $\Omega$	1
Resistencia de 300 $\Omega$	1
Resistencias de 470 $\Omega$	8
Resistencias de 1K $\Omega$	4
<b>Resistencias variables</b>	
Resistencia variable de 10K $\Omega$	1
Resistencia variable de 100K $\Omega$	1
Resistencia variable de 50K $\Omega$	1
Resistencia variable de 500K $\Omega$	1
<b>Condensadores</b>	
Condensador de 100 nF	1
Condensador de 1nF	1
<b>Diodos Led</b>	
Leds rojos	8
Leds verde	10
Led amarillo	8
<b>Displays</b>	
Displays 7 segmentos	4
Display LCD 4filas x 16caracteres	1

<b>Oscilador</b>	
Oscilador de cuarzo 1 Mhz	1
<b>Transistores</b>	
Transistores PNP	3
<b>Microinterruptores</b>	
Interruptores DIL 8 vías montaje superficial	2
<b>Circuitos integrados</b>	
Temporizadores analógicos tipo 555	2
CPLD Altera EPM7128SLC84	1
Zócalo tipo PLDD montaje superficial 84 vías	1
<b>Componentes Pasivos</b>	
Pulsadores táctiles	20
Teclado matricial de membrana	1
<b>Conectores y cables</b>	
Bornes de conexión para alimentación	1
Conector 5x2 macho	1
Terminales de pines rectos dobles	5
Terminales de pines rectos simples	5
Contacto de crimpados	80
Cable flexible	1

**Tabla 5:** *Tabla de componentes utilizados*

## **8 CONCLUSIÓN**

La ventaja que ofrece la tarjeta es que hasta ahora el alumno debía suponer que los ejercicios que realizaba estaban bien sin tener oportunidad alguna de demostrarlo.

Por esta razón la tarjeta de evaluación y prácticas para dispositivos CPLD, resulta práctica a la hora de hacer que estos demuestren sus conocimientos realizando ejercicios en ella, ya que además nos enseña a programar un dispositivo de estas características, cosa que sin esta tarjeta no podríamos llegar a conocer si no es que en un futuro el alumno se dedicaba a ello.

Las prácticas también pueden ser útiles a la hora de estudiar los elementos que conforman la tarjeta, a parte de ayudarnos a practicar con el CPLD

Otra ventaja es que se puede enfocar como un elemento de desarrollo útil para las empresas, ya que poder demostrar que un programa funciona de una manera rápida y económica es un punto que el mundo laboral agradece.

Además de estas ventajas se puede añadir el coste de fabricación de la tarjeta que ronda los 149€, es un precio que se puede asumir gracias a los beneficios que aporta, ya que ahorra tiempo a la hora de desarrollar nuevos programas y ayuda a los alumnos a conocer mejor el dispositivo.

Evidentemente el inconveniente que tiene la tarjeta de evaluación es su diseño. Ya que a la hora de evaluar nuevos proyectos estamos limitados a utilizar los elementos que hay en ella. Este inconveniente puede ser solucionado en una próxima mejora de la tarjeta de evaluación de dispositivos CPLD añadiendo más elementos de entrada y salida a ésta, o pudiendo hacer que se pueda conectar a otros elementos añadiendo terminales de entrada y salida.

En el fondo la tarjeta utilizada es un elemento que facilita: el estudio de los dispositivos programables complejos, ayuda a evaluarlos y es asequible de fabricar.



## 9 PRESUPUESTO

### 9.1 Costes del material

Descripción del material	Cantidad	Precio unidad	Total
<b>PCB</b>			
Placa fotosensibilizada para PCB (200x300)	1	16,77 €	16,77 €
<b>Resistencias</b>			
Integrados de 9 resistencias conectadas 1 K $\Omega$	3	0,31€	0,93€
Integrados de 9 resistencias conectadas 300 $\Omega$	3	0,22€	0,66€
Resistencia de 51 K $\Omega$	1	0,07€	0,07€
Resistencia de 5K1 $\Omega$	1	0,07€	0,07€
Resistencia de 300 $\Omega$	1	0,04€	0,04€
Resistencias de 470 $\Omega$	8	0,09€	0,72€
Resistencias de 1K $\Omega$	4	0,07€	0,28€
<b>Resistencias variables</b>			
Resistencia variable de 10K $\Omega$	1	0,31€	0,31€
Resistencia variable de 100K $\Omega$	1	0,32€	0,32€
Resistencia variable de 50K $\Omega$	1	0,29€	0,29€
Resistencia variable de 500K $\Omega$	1	0,29€	0,29€
<b>Condensadores</b>			
Condensador de 100 nF	1	0,16€	0,16€
Condensador de 1nF	1	0,16€	0,16€
<b>Diodos Led</b>			
Leds rojos	8	0,16€	1,28€
Leds verde	10	0,16€	1,6€
Led amarillo	8	0,16€	1,28€

<b>Displays</b>			
Displays 7 segmentos	4	1,58€	6,32€
Display LCD 4filas x 16caracteres	1	23,5€	23,5€
<b>Oscilador</b>			
Oscilador de cuarzo 1 Mhz	1	2,66€	2,66€
<b>Transistores</b>			
Transistores PNP	3	0,466€	1,398€
<b>Microinterruptores</b>			
Interruptores DIL 8 vías montaje superficial	2	1,9€	3,8€
<b>Circuitos integrados</b>			
Temporizadores analógicos tipo 555	2	0,406€	0,812€
CPLD Altera EPM7128SLC84	1	16,42€	16,42€
Zócalo tipo PLDD montaje superficial 84 vías	1	27,3€	27,3€
<b>Componentes Pasivos</b>			
Pulsadores táctiles	20	0,28€	5,6€
Teclado matricial de membrana	1	6,09€	6,09€
<b>Conectores y cables</b>			
Bornes de conexión para alimentación	1	0,55€	0,55€
Conector 5x2 macho	1	0,34€	0,34€
Terminales de pines rectos dobles	5	1,62€	8,1€
Terminales de pines rectos simples	5	0,79€	3,95€
Contacto de encaste	80	0,05€	4€
Cable flexible	1	6,7€	6,7€
<b>Costes del material</b>			<b>142,77 €</b>

Tabla 6: Tabla de costes del material

## 9.2 Costes relacionados con la ingeniería

Concepto	Horas	Precio/hora	Total
Estudio y documentación	80	45 €	3.600 €
Diseño	80	45 €	3.600 €
montaje	40	25 €	1.000 €
Redacción de la memoria	40	45 €	1.800 €
<b>Total</b>			<b>10.000 €</b>

Tabla 7: costes relacionados con la ingeniería

## 9.3 Amortización del instrumental

Equipo utilizado	horas de utilización	Precio/hora	Total
Ordenador PC	120	5€	600€
Software Orcad Ver.10.0	80	8€	640€
Software Max Plus II	40	8€	320€
Micr. Word Xp	40	2€	80€
<b>Total</b>			<b>1640€</b>

Tabla 8: Amortización del instrumental

## 9.4 Coste de fabricación del prototipo

<b>Costes del material</b>			142,77 €
<b>Costes de Ingeniería</b>			10.000 €
<b>Costes del Equipo</b>			1640€
<b>Total</b>			<b>11782,77€</b>

Tabla 9: Coste de fabricación del prototipo

## 9.5 Coste de fabricación

<b>Costes de fabricación de 1 unidad.</b>	<b>35 €</b>
---	-------------

Tabla 10: Coste de fabricación

### 9.6 Estudio económico para la fabricación de 100 Uds.

Costes	Ecuación para 100Uds	Total
Coste total del material (por 100 Uds.)	$(142,77 - 20\%)*100$	11.421,60 €
Costes de fabricación (por 100 Uds.)	$35*100$	3.500 €
Coste total para 100 unidades		14.921,60 €
<b>Coste por unidad</b>		<b>149 €</b>

Tabla 11: Estudio económico para 100UDs.

Para la fabricación de 100 unidades se ha tenido en cuenta que a la hora de comprar grandes cantidades nos harán un precio más asequible por los materiales. Por eso hemos descontado un 20%.

El coste por unidad será de 149 €, pero si lo que se quiere es venderlo en el mercado, le tenemos que añadir el coste de la fabricación del prototipo.

### 9.7 Estudio del precio de venta en el mercado

Precio en el mercado	
Concepto	Total
Coste total del material (por 100 Uds.)	11.421,60 €
Costes de fabricación (por 100 Uds.)	3.500 €
Coste del prototipo	11.542,77 €
Total	26.464,37 €
Precio unitario	264,64 €
<b>Precio con un beneficio del 50%</b>	<b>396,97 €</b>

Tabla 12: Estudio del precio de venta

La tarjeta de evaluación y prácticas para dispositivos CPLD se podría vender en el mercado a un precio de 390€.

Incluso el precio de venta podría ser menor dependiendo del beneficio que queramos obtener, llegando a valer en el mercado 273€ si el beneficio que queremos es del 5%

## 10 ESQUEMAS

Los esquemas que se muestran en este apartado aparecen en el anexo esquemas del CD que se adjunta junto con esta memoria.

### 10.1 Esquemas de conexionado de los módulos

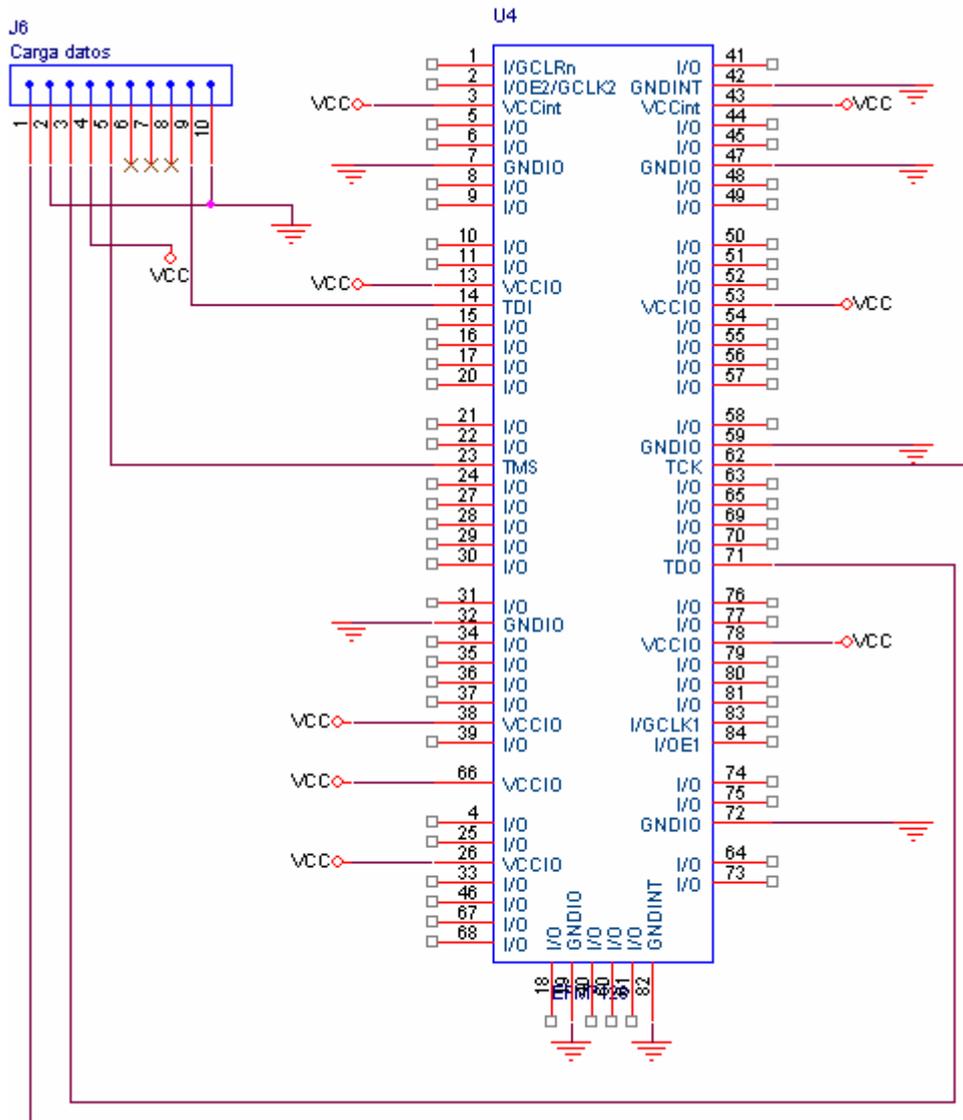


Figura 26: Esquema de conexionado del CPLD

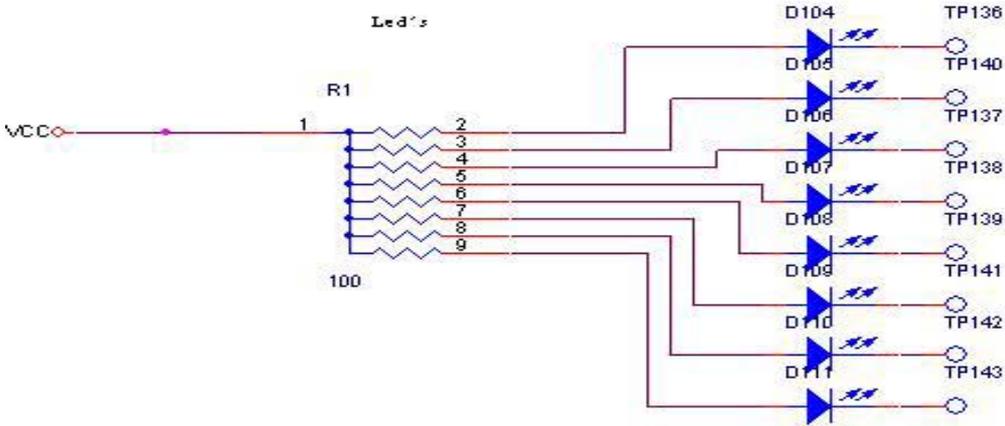


Figura 27: Conexionado de los diferentes Leds

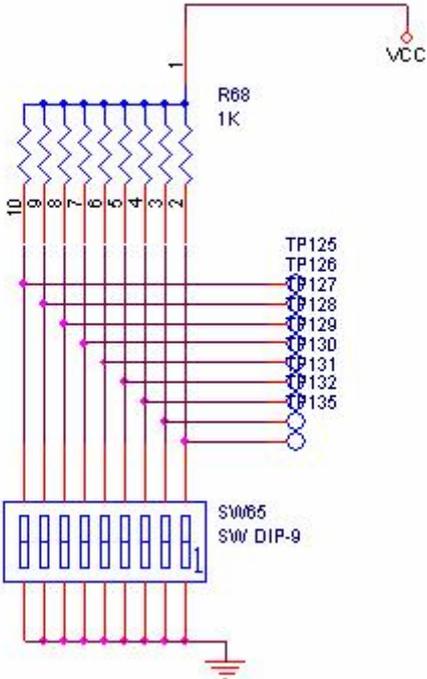


Figura 28: Conexionado del interruptor tipo DIL o microrruptor



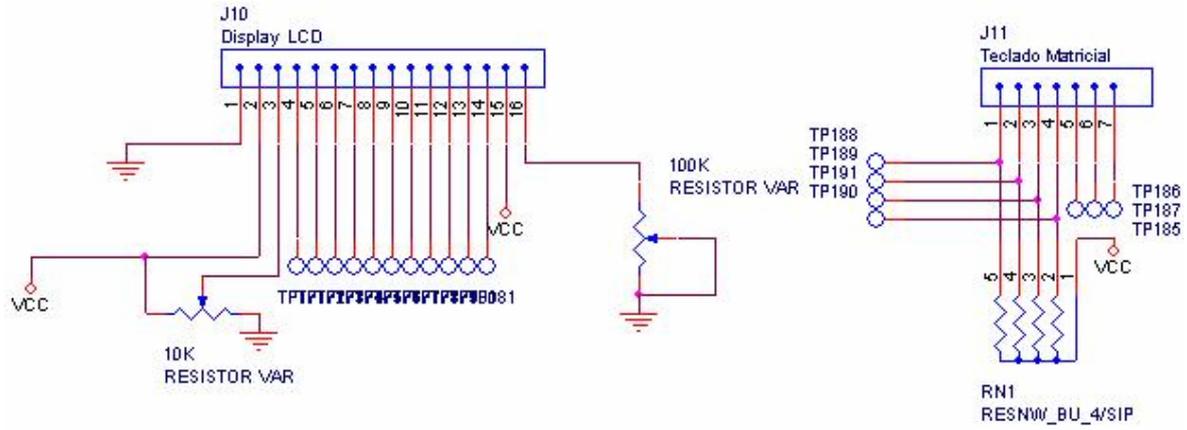


Figura 31: Esquema conexión Display LCD y teclado matricial respectivamente

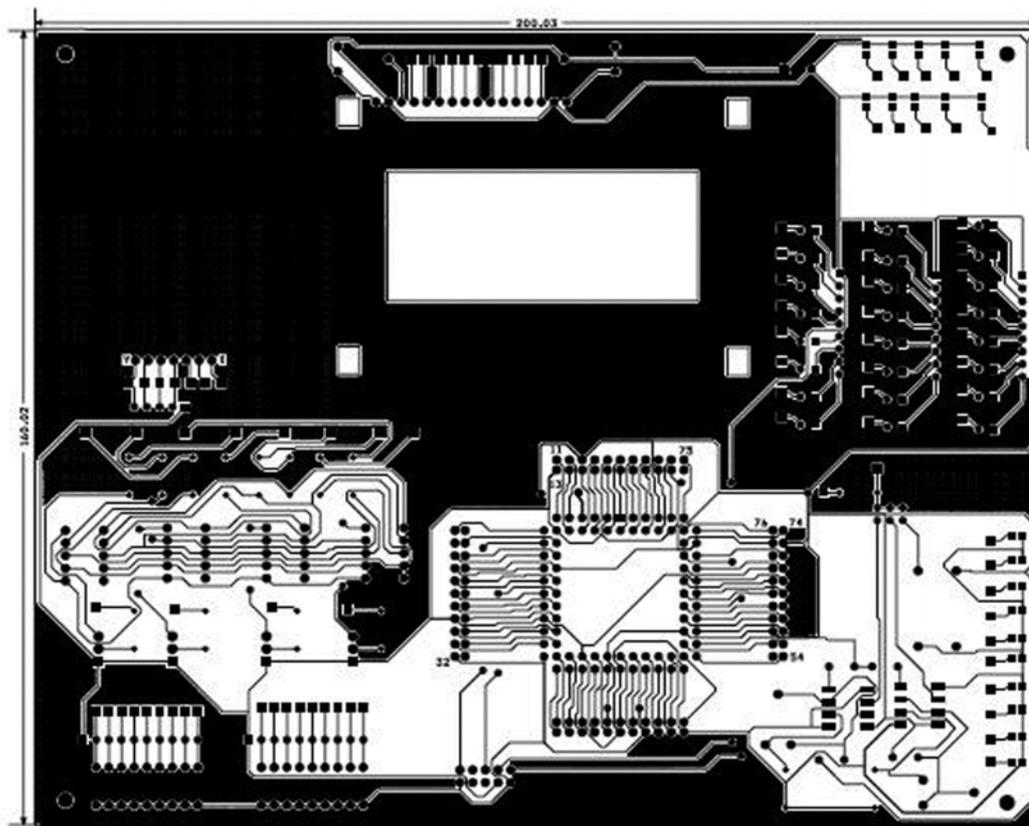


Figura 32: Esquema circuito integrado implementado en la tarjeta

## **11 ÍNDICE DE FIGURAS**

Figura 1: Diagrama de bloques de un SPLD.....	6
Figura 2: Composición de un bloque lógico.....	6
Figura 3: Diagrama de programación del bloque lógico de una PROM.....	8
Figura 4: Diagrama de programación del bloque lógico de una PAL.....	9
Figura 5: Diagrama de programación del bloque lógico de una FPLA.....	10
Figura 6: Ejemplo de la arquitectura interna de un CPLD.....	11
Figura 7: Diagrama de bloques de un FPGA.....	13
Figura 8: Diagrama de bloques de un CPLD de la familia MAX.....	16
Figura 9: Diagrama de funcionamiento de una macrocelda.....	17
Figura 10: Diagrama de funcionamiento del bloque de entrada/salida.....	19
Figura 11: Diagrama de funcionamiento de la PIA.....	20
Figura 12: Esquema de los diferentes módulos que conforman la tarjeta de evaluación... 23	23
Figura 13: Encapsulado utilizado por la tarjeta de evaluación del CPLD, y numeración de los terminales.....	26
Figura 14: Esquema de los segmentos que conforman un Led 7 segmentos.....	32
Figura 15: Diagrama de funcionamiento de un teclado matricial.....	37
Figura 16 Señal de barrido de columnas.....	38
Figura 17: Ejemplo de la pulsación de una tecla.....	39
Figura 18: Diagrama de conexión de un cristal de cuarzo de 4 terminales.....	42
Figura 19: Descripción de los terminales del 555.....	43
Figura 20: Esquema de conexión del 555.....	44
Figura 21: En este diagrama observamos los tiempos a los que hacemos referencia.....	44
Figura 22: Ejemplo del generador de barrido.....	55
Figura 23: Ejemplo del funcionamiento de un contador.....	55
Figura 24: Tabla de interpretación de valores del teclado.....	56
Figura 25: Esquema funcionamiento de la maquina de estados del teclado.....	57
Figura 26: Esquema de conexión del CPLD.....	67
Figura 27: Conexión de los diferentes Leds.....	68
Figura 28: Conexión del interruptor tipo DIL o microinterruptor.....	68
Figura 29: Esquema de montaje del display 7 segmentos.....	69
Figura 30: Esquema montaje osciladores.....	69
Figura 31: Esquema conexión Display LCD y teclado matricial respectivamente.....	70
Figura 32: Esquema circuito integrado implementado en la tarjeta.....	70



## **12 ÍNDICE DE TABLAS**

Tabla 1: <i>Numeración de los pins de programación</i> .....	27
Tabla 2: <i>Numeración de los pins del reloj</i> .....	27
Tabla 3: <i>Numeración de los pins de habilitación</i> .....	27
Tabla 4: <i>Numeración de los pins de alimentación</i> .....	28
Tabla 5: <i>Tabla de componentes utilizados</i> .....	60
Tabla 6: <i>Tabla de costes del material</i> .....	64
Tabla 7: <i>costes relacionados con la ingeniería</i> .....	65
Tabla 8: <i>Amortización del instrumental</i> .....	65
Tabla 9: <i>Coste de fabricación del prototipo</i> .....	65
Tabla 10: <i>Coste de fabricación</i> .....	65
Tabla 11: <i>Estudio económico para 100UDs.</i> .....	66
Tabla 12: <i>Estudio del precio de venta</i> .....	66



## 13 BIBLIOGRAFÍA

- [1] **Enrique Maldonado, L. Jacobo, M<sup>a</sup> Dolores Valdés.** *Dispositivos Lógicos Programables y sus aplicaciones*, Ed. Paraninfo, 2002
- [2] **J.M<sup>a</sup> García Iglesias, Emilio J. Pérez.** *Dispositivos Lógicos Programables. (PLDs). Diseño práctico de aplicaciones*, Ed. RA-MA, 2006
- [3] **Ángel Bueno Martín, Ana de Soto Gorroño.** *Desarrollo y construcción de prototipos Electrónicos*, Ed. Marcombo, 2005
- [4] **Daniel Bosch Castellà,** *Tarja d'avaluació d'EPLD's, Volum I*, setembre 1997
- [5] <http://www.altera.com/literature/ds/m7000.pdf>, *Max 7000 Programmable Logic Device*.
- [6] [www.rsonline.es](http://www.rsonline.es), *RS-El catalogo*, Octubre 2006-2007