

## Unidad IV: Lenguajes HDL

### 4.1. Dispositivos lógicos programables

Lógica programada es un tipo de diseño implementado en chips que permite la reconfiguración de los circuitos con el simple cambio del software que incorpora, es lo contrario de la lógica cableada.

La lógica programada se basa en dispositivos lógicos programables (*PLD*), los cuales tienen una función no establecida, al contrario que las puertas lógicas que tienen una función fija en el momento de su fabricación. Antes de poder utilizar el PLD en un circuito, este debe ser programado.

#### 4.1.1. Tipos

##### **PAL**

MMI introdujo un dispositivo revolucionario en 1978, la Programmable Array Logic (*Matriz lógica programable*). La arquitectura era más sencilla que la FPLA de Signetics porque omitía la matriz OR programable. Esto hizo los dispositivos más rápidos, más pequeños y más baratos. Estaban disponibles en encapsulados de 20 pines y DIP de 300 milésimas de pulgada, mientras que las FPLAs venían en encapsulados de 28 pines y DIP de 600 milésimas de pulgada. Ciertas publicaciones sobre PALs desmitificaban el proceso de diseño. El software de diseño PALASM (*PAL Assembler*, ensamblador PAL) convertía las ecuaciones Booleanas de los ingenieros en el patrón de fusibles requerido para programar el dispositivo. Los PAL de MMI pronto fueron distribuidos por National Semiconductor, Texas Instruments y AMD.

##### **GALs**

Una innovación del PAL fue la *matriz lógica genérica* (*Generic array logic*) o GAL. Ambas fueron desarrolladas por Lattice Semiconductor en 1985. Este dispositivo

tiene las mismas propiedades lógicas que el PAL, pero puede ser borrado y reprogramado. La GAL es muy útil en la fase de prototipado de un diseño, cuando un fallo en la lógica puede ser corregido por reprogramación. Las GALs se programan y reprograman utilizando un programador OPAL, o utilizando la técnica de programación circuital en chips secundarios.

## FPGAs

Mientras el desarrollo de las PALs se enfocaba hacia las GALs y CPLDs (ver secciones superiores), apareció una corriente de desarrollo distinta. Esta corriente de desarrollo desembocó en un dispositivo basado en la tecnología de matriz de puertas y se le denominó field-programmable gate array (FPGA). Algunos ejemplos de las primeras FPGAs son la matriz 82s100 y el secuenciador 82S105 de Signetics, presentados a finales de los 70. El 82S100 era una matriz de términos AND, y también tenía funciones de biestable.

Las FPGAs utilizan una rejilla de puertas lógicas, similar a la de una matriz de puertas ordinarias, pero la programación en este caso la realiza el cliente, no el fabricante. El término field-programmable (literamente programable en el campo) se refiere a que la matriz se define fuera de la fábrica, o "en el campo".

Las FPGAs se programan normalmente tras ser soldadas en la placa, en una forma similar a los CPLDs grandes. En las FPGAs más grandes, la configuración es volátil y debe ser reescrita cada vez que se enciende o se necesita una funcionalidad diferente. La configuración se guarda normalmente en una PROM o EEPROM. Las versiones EEPROM pueden ser programadas mediante técnicas como el uso de cables JTAG.

### **4.1.2. Características**

### **4.1.3. Fabricantes**

#### **4.1.4. Pasos para el diseño con PLD's**

Son IC's estándar de la familia de ASIC's que están disponibles en configuraciones estándar desde catálogos de partes y se venden en grandes volúmenes a muchos consumidores. Sin embargo, los PLD's pueden configurarse o programarse para crear partes configurables para una aplicación específica, los PLD's utilizan diferentes tecnologías para permitir la programación del dispositivo. Entre las principales características de los PLD's se puede destacar:

- No poseen mascararas o capas ni celdas lógicas configurables.
- Rápido diseño.
- Un solo gran bloque de interconexiones programables.
- Poseen una matriz de macroceldas lógicas que usualmente consiste de un arreglo programable lógico seguido por un flip-flop o latch.

### **4.2. Programación de circuitos combinacionales con HDL**

#### **4.2.1. Por captura esquemática**

#### **4.2.2. Por tabla de verdad**

#### **4.2.3. Por ecuaciones booleanas**

#### **4.2.4. Por descripción de comportamiento**

### **4.3. Programación de circuitos secuenciales con HDL**

#### **4.3.1. Por captura esquemática**

#### **4.3.2. Por tabla de verdad**

#### **4.3.3. Por ecuaciones booleanas**

**4.3.4. Por descripción de comportamiento**

**4.3.5. Por tabla de estado**

**4.3.6. Por diagrama de transición**