

Arquitectura de los PLD´s y sus aplicaciones.

PROGRAMACIÓN DE SISTEMAS EMBEBIDOS.



24 de marzo de 2020

nadia sarahi murguia chavez

ING. MECATRONICA 8vo A

**Definición de PLD.**

Los Dispositivos lógicos Programables (PLD´s) son dispositivos programables por el usuario para aplicar gran variedad de funciones lógicas en diferentes sistemas.

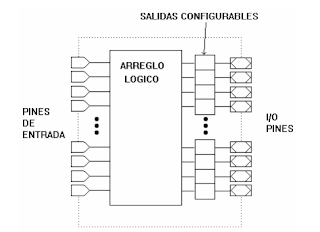


Ilustración 1.PLD´s

**Arquitectura básica de un PLD**

Las entradas vienen desde un arreglo lógico, estos arreglos están hechos por filas y columnas, como se muestra en la siguiente ilustración.

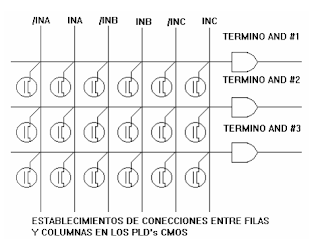


Ilustración 2. Entradas de un PLD.

Cada par de columnas representa la entrada negada o complementada y la misma entrada sin negar y cada fila representa un término AND. Las conexiones lógicas se establecen entre diferentes columnas y filas en el arreglo para determinar cual combinación de entradas llevara al termino AND a un nivel alto.

Más de un término AND alimenta una compuerta OR. La salida es la suma de productos.

**DIFERENTES TIPOS DE PLD's**

Los diferentes tipos de PLD´s son los PAL´s (Programmable Array Logic) o Lógica en un Arreglo Programable en español. El otro tipo de PLD es el PLA (Programmable Logic Array) o Arreglo Lógico Programable en español.

**LENGUAJES DE PROGRAMACIÓN DE LOS PLD**

Actualmente cada fabricante de estos dispositivos diseña su propio software para programar dichos dispositivos lo que quiere decir que existen una gran variedad de lenguajes de programación ya que son demasiado fabricantes. Los lenguajes más populares a nivel mundial son 3:

* Lenguaje de contactos o Ladder
* Lenguaje Booleano (Lista de instrucciones)
* Diagrama de funciones}

**Ejemplos de aplicaciones con PLD**

* Conversores de código (por ejemplo, de binario a gray o viceversa)
* Conversores de binario a 7 segmentos (que también muestren las letras A-F)
* Verificadores de paridad y chequeo de errores
* Distintos tipos de contadores y registros
* Controladores de memoria y E/S para microprocesadores
* Tablas predefinidas (LUT)