



## Mapa Conceptual

**Nombre del alumno:** Yahir Aguilar Sicalhua.

**Nombre del tema:** Bloques funcionales secuenciales y circuitos programables.

**Parcial:** 1.

**Nombre de la materia:** Diseño Lógico.

**Nombre del profesor:** Juan José Ojeda Trujillo.

**Nombre de la licenciatura:** Ingeniería en Sistemas Computacionales.

**Cuatrimestre:** 6.

# Unidad III. Bloques Funcionales Secuenciales.

## 3.1.- Contadores

El PLC Simatic S7-200 ofrece al usuario un conjunto de instrucciones que permiten llevar la cuenta de cuántos eventos se producen durante la ejecución del programa de control, esto es, cuántas veces una señal (una marca, una entrada, una variable, etc.) cambia de valor.

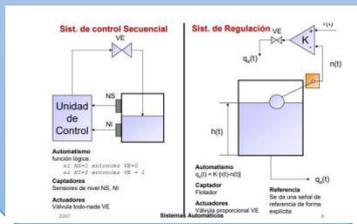
## 3.2.- Registros de desplazamiento.

Los registros de desplazamiento son circuitos secuenciales formados por biestables o flip-flops generalmente de tipo D conectados en serie y una circuitería adicional que controlará la manera de cargar y acceder a los datos que se almacenan.

## 3.3.- Registros conectados en anillo.

El MRP sirve para aumentar la disponibilidad de la red con topología en anillo y para tener una comunicación libre de bucles.

## 3.4.- Representación comportamental de bloques funcionales secuencial mediante VHDL.



## 3.5.- Implementación de sistemas secuenciales síncronos mediante bloques funcionales secuenciales y puertas lógicas

```
Exemplo Descrito en VHDL: un contador síncrono con reset asincrónico y carga en paralelo (load).
* Función de funcionamiento:
- EJEMPLO DE UN CONTADOR CON EVENTOS
LIBRARY IEEE;
USE IEEE.STD_LOGIC_Types;
USE IEEE.STD_LOGIC_ARNTHALL;
USE WORK.STD_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY contador IS
PORT (
    clk: IN STD_LOGIC;
    reset: IN STD_LOGIC;
    load: IN STD_LOGIC;
    count: OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
);
END ENTITY;
ARCHITECTURE ejemplo de contador IS
BEGIN
    PROCESS (clk, reset, load, exp) BEGIN
        IF reset='1' THEN
            count <= (0);
        ELSIF (clk'EVENT AND clk='1') THEN
            IF load='1' AND exp='1' THEN
                count <= exp;
            ELSIF load='0' AND exp='0' THEN
                count <= count + 1;
            ELSE
                count <= count + 1;
            END IF;
        END IF;
    END PROCESS;
END ejemplo;
```

## 3.6.- Representación estructural de la implementación del sistema, basada en bloques funcionales secuenciales, mediante VHDL.



## 3.7.- Memorias no volátiles.

Son dispositivos electrónicos enchufables en la CPU, destinados a guardar información de manera provisional o permanente. Se cuentan con dos tipos de memorias, volátiles (RAM) y no volátiles (EPROM Y EEPROM), según requieran o no de energía eléctrica para la conservación de la información.

## 3.8.- Tipos de memorias no volátiles.

- Memorias EEPROM
- Memorias FLASH
- Memorias EERAM

## 3.9.- Implementación de circuitos combinacionales mediante memorias no volátiles.

Todos los PLDs están formados por matrices programables. Una matriz programable es una red de conductores distribuidos por filas y columnas con un fusible en cada punto de intersección.

- Matriz OR.
- Matriz AND.

## 3.10.- Representación comportamental de memorias no volátiles mediante VHDL.

Memoria programable de sólo lectura (Programmable Read Only Memory, PROM). Está formada por un conjunto fijo de puertas AND (no programable) conectadas como decodificador y

## 3.11.- Implementación de sistemas secuenciales síncronos mediante bloques funcionales secuenciales y memorias no volátiles.

Las operaciones básicas de una memoria son las de escritura y lectura. La operación de escritura coloca los datos en una posición específica de la memoria y la operación de lectura extrae los datos de una posición específica de la memoria.

## 3.12.- Representación estructural de la implementación del sistema, basado en memorias no volátiles, mediante VHDL.

Estas memorias son conocidas habitualmente como caches, pero normalmente no son realmente caches, sino memorias de tipo SRAM controladas por software (es decir, scratchpads).

## Unidad IV. Circuitos Programables.

### 4.1.- Circuitos full custom y semicustom.

Los diseños electrónicos tienen que cumplir varios requisitos a nivel de prestaciones, consumo y área. Cuando trabajamos con diseños sencillos se utilizan componentes con encapsulados grandes de agujero pasante (DIP).

### 4.2.- Tipos de circuitos lógicos programables: Standard Cell, PLA/PAL, CPLD y FPGA.

Las celdas estándar están diseñadas en función de la potencia, el área y el rendimiento. El primer paso es la arquitectura celular. La arquitectura de la celda se trata de decidir la altura de la celda en función de los requisitos de tono y biblioteca.

### 4.3.- Implementación de circuitos combinacionales mediante circuitos lógicos programables de tipo PLA y PAL.

Un arreglo lógico programable (PLA) realiza la misma función que una ROM. Un PLA con  $n$  entradas y  $m$  salidas puede realizar  $m$  funciones de  $n$  variables. La organización interna del PLA difiere de la de la ROM, el decodificador se reemplaza por un arreglo de ANDs que realiza los términos producto seleccionados de las variables de entrada.

### 4.4.- Dispositivos lógicos programables/ Generic Array Logic

Conforme avanzaba la tecnología de circuitos integrados, una gran variedad de dispositivos lógicos programables apareció. Los PALs tradicionales no son reprogramables, sin embargo, existen ahora PALs borrables y reprogramables con tecnología flash. A veces, a éstos se les llama PLDs.

Fuente de información:

<https://plataformaeducativauds.com.mx/libro.php?idLibro=168660366316>