

# Práctica 6: Diseño de circuitos con dispositivos lógicos programables

## Objetivos:

- Analizar el funcionamiento de los distintos tipos de dispositivos lógicos programables: memorias ROM, PAL, PLDs, CPLDs, etc.
- Diseñar circuitos con dispositivos lógicos programables
- Utilizar herramientas de CAD para la programación de los dispositivos lógicos.
- Montar y comprobar sistemas con dispositivos lógicos programables

NOTA: salvo que se indique lo contrario, los circuitos podrán realizarse con GAL16V8, GAL22V10, GAL26V12, alguna EPROM, Lattice ispMACH4A532/32, Xilinx XC9536 usándose el menor dispositivo posible

## Actividad 1: Realización de un decodificador binario a 7 segmentos

1. Con una PAL16V8 o 22V10 (el menor dispositivo posible), realizar un decodificador BCD a 7 segmentos con entradas activas a nivel alto y salidas activas a nivel bajo y que represente los números del 0 al 9 y las letras de la A a la F correspondientes a las 15 combinaciones binarias.
2. Probar el circuito diseñado, comprobando su funcionamiento en distintos casos.

(se sugiere usar una tabla en ABEL).

## Actividad 2 : realización de un comparador

1. Realizar un comparador de dos números de 3 bits, con salidas Mayor, Igual y Menor.
2. Probar el circuito diseñado, comprobando su funcionamiento en distintos casos.
3. (opcional) modificar el circuito para hacer un comparador de dos números de 8 bits

(se sugiere usar ecuaciones en ABEL)

## Actividad 3 (opcional): Realización de un detector de paridad

1. Diseñar un generador/detector de paridad de números de 8 bits, seleccionando el PLD que se crea conveniente, intentando utilizar el más sencillo posible de todos los disponibles.
2. Probar el circuito con distintas combinaciones de entrada comprobando su funcionamiento.
3. (opcional) Realizar el circuito con una EPROM, y verificar su funcionamiento

(se sugiere usar esquemáticos o ecuaciones ABEL)

## Actividad 4 (opcional): Realización de un contador personalizable

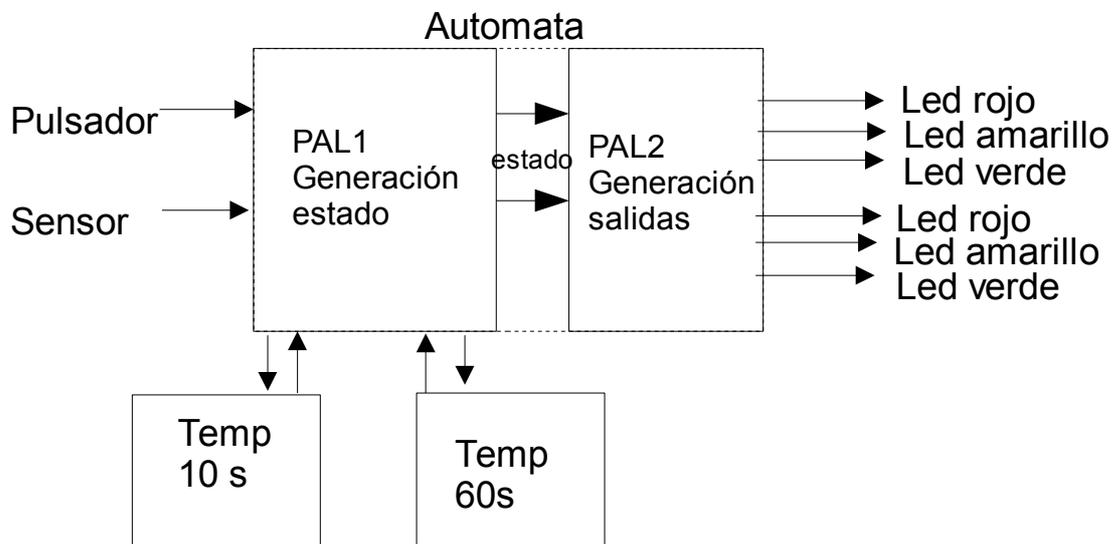
1. El contador tendrá dos entradas de cuatro bits: Inicio[3..0] y Fin[3..0] y la salida de cuenta Q[3..0]. El contador contará al ritmo marcado por la señal de reloj, empezando por Inicio y dará la vuelta cuando llegue a Fin.
2. Probar el circuito para distintas combinaciones de Inicio y Fin.
3. (opcional) modificar el circuito para que el contador incorpore una entrada UD y otra HAB. La entrada UD será Up/Down (con 0 contará para arriba) y la HAB será la de habilitación de cuenta (con 0 el contador estará parado).

(se sugiere usar ecuaciones)

## Actividad 5: Implementación de un autómata con PLD

Se va a implementar el autómata para el control de un cruce con entre dos calles de una sola dirección (una principal con bastante tráfico y una secundaria con poco tráfico) regulada por semáforos, con las siguientes características:

- La vía principal tiene un semáforo, que estará habitualmente en verde para los coches.
- La vía secundaria tiene un semáforo que estará habitualmente en rojo para los coches.
- La vía principal está atravesada por un paso de peatones.
- Cuando un peatón quiere cruzar, accionará un pulsador, pasados 60s de la pulsación el semáforo de la vía principal pasará a ámbar, y pasados 10s más, pasará a rojo, pasando el de la vía secundaria a verde. Permanecerán así durante 60s, transcurridos los cuales, el semáforo de la vía secundaria pasará a ámbar y 10s después pasará a rojo, a la vez que el semáforo de la vía principal pasa a verde.
- Cuando un coche se para delante del semáforo en la vía secundaria, un sensor lo detectará y se activará. Cuando el sensor se active, ocurrirá una secuencia igual que la descrita anteriormente, pero con otra temporización: inmediatamente, el semáforo principal se pone en ámbar, a los 10s pasan a SP=R SS=V, en otros 10s pasan a SP=R SS=A, en otros 10s pasan a SP=V SS=R.



Para la realización dispondremos de un reloj de 1s (habrá que diseñarlo a partir de un aestado). Cada temporizador se realizará con una PAL16V8. El autómata (que será de MOORE) se realizará con dos PAL16V8, una de ellas (secuencial) servirá para generar el estado, se programará mediante diagrama de estados o bien tabla registrada, la otra (combinacional) generará la salida a partir del estado, se programará mediante una tabla combinacional.

1. Diseñar los temporizadores de forma que tengan una salida FINDECUENTA y una entrada INICIALIZACION, activas a nivel alto o bajo, según se decida.
2. Realizar la programación del autómata mediante el diagrama de estados con Lattice ispLEVER.
3. Seleccionar un dispositivo y programarlo.
4. Simular el funcionamiento del circuito con Proteus
5. Montar el circuito en una placa y comprobar su funcionamiento.

La memoria deberá incluir, como mínimo:

- Esquemas de todos los circuitos montados.
- Funciones desarrolladas, con su justificación, simplificación, etc.
- Diagramas de estados desarrollados.
- Listados de la programación de cada dispositivo.
- Observaciones y comentarios