Práctica 4: 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, Lattice ispMACH4A532/64 , Xilinx XC9536 Xilinx XC9572, usándose el menor dispositivo posible

La información sobre la plataforma PLUMABOT se encuentra en www.plumabot.es

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

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. Simular el funcionamiento del circuito con proteus conectándolo las entradas del codificador a un contador binario reversible de 4 bits
3. Probar el circuito diseñado, comprobando su funcionamiento de la misma forma que en proteus.

(se sugiere usar una tabla en ABEL).

Actividad 2 : Realización de un sumador

1. Realizar un sumador binario de dos números de 4 bits
    - Utilizando esquemáticos
    - Utilizando ecuaciones
2. Probar ambos circuitos en proteus comprobando su funcionamiento
3. Montar uno de los dos circuitos y probar su funcionamiento

Actividad 3 : Realización de un codificador con prioridad

1. Diseñar un codificador 8:3 prioritario con entradas y salidas activas a nivel alto, en el que se de prioridad a la entrada de menor peso.
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 una tabla en ABEL)

Actividad 4: Realización de un generador PWM

El circuito consistirá en un generador de señal PWM programable. Para ello dispondremos de una señal de reloj de 1Khz (se generará con el reloj del entrenador). El circuito tendrá una entrada de 4 bits, por la que se introducirá un número de 0 a 15 indicando el ciclo de trabajo de la señal PWM, siendo para 0000 el 0% y para 1111 el 100%.

Para conseguir el efecto deseado el circuito se diseñará de la siguiente forma: se creará un contador de 4 bits, que contará al ritmo de la señal de reloj de 0 a 15, cuando la cuenta del contador sea menos que el código introducido se activará la salida, cuando no lo sea, la salida estará desactivada.

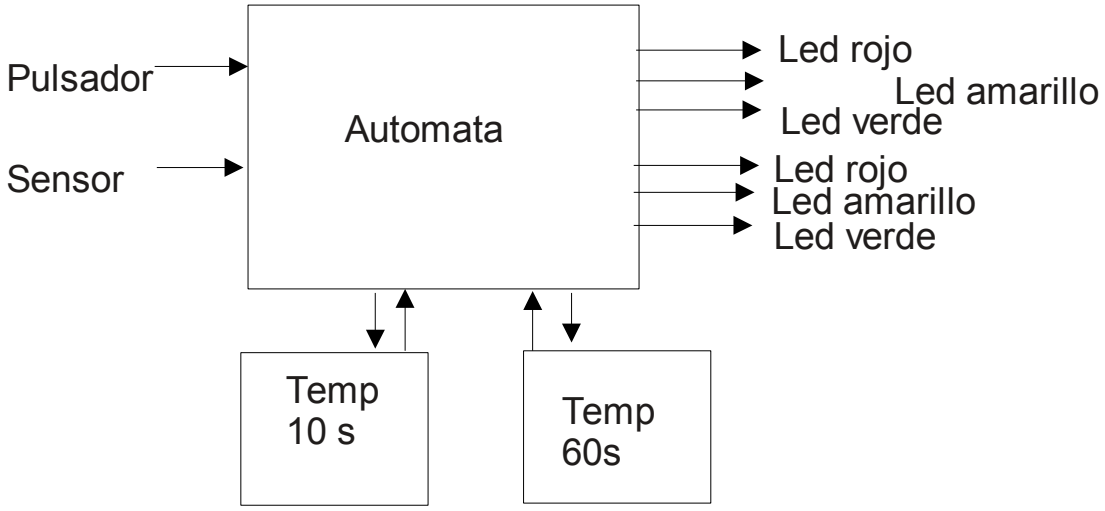
1. El circuito se diseñará con una PAL16V8 o PAL22V10, se probará con proteus, se montará y se comprobará su funcionamiento con el analizador lógico, mostrando tanto la salida como la cuenta del contador.
2. El circuito se montará en la placa base PLUMABOT-PMB03 con la placa de periféricos PEB07. Como entradas de selección se utilizaran los 4 interruptores de la placa de periféricos y como salida uno de los 8 leds. Se montará el circuito, seleccionando como frecuencia de reloj 50Khz y se medirá la señal de reloj y la salida con el analizador lógico.
3.

(se sugiere usar ecuaciones, aunque también puede hacerse con esquemas)

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 aestable). Cada temporizador se realizará con una PAL16V8. El autómata (que será de MEALY) se realizará con una gal26V12, o bien con una ispMACH4A532/32.

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. Simular el funcionamiento de cada temporizador con Proteus
3. Realizar la programación del autómata mediante el diagrama de estados con Lattice ispLEVER.
4. Seleccionar un dispositivo y programarlo.
5. Montar el circuito en una placa y comprobar su funcionamiento.
6. (opcional). Realizar el ejercicio con la plataforma PLUMABOT-PMB03 y PEB07, en cuyo caso se realizará todo junto con la XC95108