

APELLIDOS : \_\_\_\_\_

NOMBRE: \_\_\_\_\_ D.N.I.: \_\_\_\_\_

- LEER ATENTAMENTE EL ENUNCIADO.
- RAZONAR CADA UNA DE LAS DECISIONES ADOPTADAS TANTO EN EL HW COMO EN EL SW.
- CUIDAR LA PRESENTACIÓN EVITANDO ESQUEMAS ININTELIGIBLES, LETRA INDESCIFRABLE Y EXPRESIONES MALSONANTES.
- UTILIZAR PARA EL DISEÑO LOS CRITERIOS DE INGENIERÍA (MÁXIMA EFICIENCIA, MÍNIMO COSTO).
- UTILIZAR COMENTARIOS EN LA ELABORACIÓN DEL CÓDIGO SW.

P1. Diseñar un controlador muy básico de una lavadora con las siguientes especificaciones:

ENTRADAS:

- Interruptor de marcha, 1 comienzo de lavado, 0 parada incluso en medio del ciclo de lavado (D).
- Sensor de que el tambor está lleno de agua (A) o está vacío.
- Sensor de que el tiempo de lavado se ha agotado (T).

SALIDAS:

- Bomba de agua de entrada (B1).
- Bomba de agua de salida (B2).
- Motor de lavado en marcha y puesta a cero de tiempo (M1).
- Motor de centrifugado en marcha y puesta a cero de tiempo (M2).

Cuando el controlador recibe el pulsador de marcha D, la lavadora se llena de agua y efectúa el lavado hasta que el temporizador alcanza cero activando la señal de tiempo agotado T. A continuación vacía con la bomba de agua B2 el agua sucia y llena el tambor con agua limpia y fría y vuelve a activar el motor M1 hasta que se vuelve a alcanzar el tiempo. Finalmente, tras vaciar el agua del aclarado, realiza el centrifugado con la ropa seca activando el motor M2, completando el ciclo.

Simplificar el diseño ignorando los estados imposibles en la entrada, dibujar el diagrama de estados (2 puntos) y emplear cualquier tipo de biestable o bloque secuencial para su implementación (2 puntos).

P2. Teniendo en cuenta la interfaz con el 8051, diseñar la conexión de un registro de solo escritura y otro de solo lectura, ambos en la dirección 0FFF0H del mapa de memoria externa. Utilizar registros LS374, decodificador LS138 y puertas lógicas.

No es necesario decodificar totalmente el bus de direcciones, la zona libre de memoria de datos externa (xdata) comienza en 8000H y termina en FFFFH.

- a) Especificar el diagrama esquemático del control y conexión de ambos registros. (2 puntos)
- b) Especificar razonadamente el mapa de memoria resultante tras la implementación anterior para la zona de datos externa. (1 punto)

P3. Un teclado ASCII emite el código ASCII en paralelo y la paridad al puerto P3 de un sistema 8051. El teclado también genera una señal de 'strobe' al bit P1.0.

Cuando se pulsa una tecla, el teclado saca el código ASCII de la tecla pulsada y paridad en las ocho líneas paralelas y saca un pulso de validación positivo (strobe) durante 1ms. El programa debería muestrear la señal de strobe continuamente hasta que esté a nivel alto.

Luego el programa deberá leer el código ASCII, enmascarar el bit de paridad (D7), y almacenar este código ASCII en una tabla en memoria. Luego se volverá a muestrear la señal de strobe hasta que se encuentre a nivel bajo. Cuando la señal strobe se encuentre a nivel bajo, comprobará si se han leído ya 10 caracteres. Si todavía no, volverá a esperar la subida a 1 de la señal strobe de nuevo para leer un nuevo carácter. Si ya se han leído 10 caracteres, parar.

- a) Dibuja el organigrama del proceso de lectura anterior (1 punto).
- b) Codificar en lenguaje C la función encargada de leer de teclado: void teclado(char \*x). No es necesario comprobar la paridad. El parámetro que se le pasa será la dirección de un array de caracteres, donde se almacenará la cadena (2 puntos).