

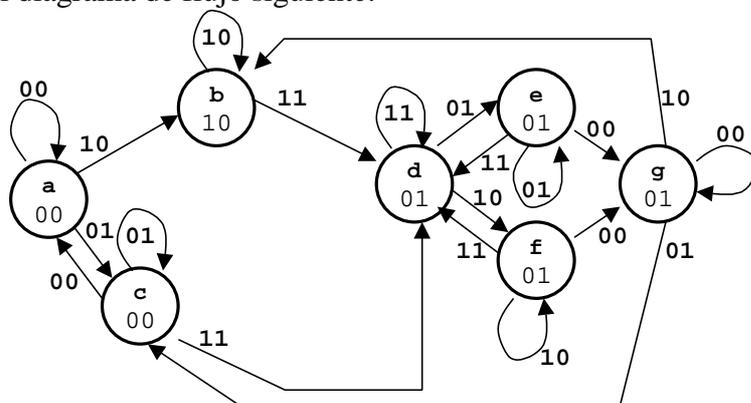
APELLIDOS : _____
 NOMBRE: _____ D.N.I.: _____

NOTAS:

- ALUMNOS CON PRIMER PARCIAL EJERCICIOS 1 Y 2
- ALUMNOS CON SEGUNDO PARCIAL EJERCICIOS 3 Y 4
- ALUMNOS CON TODO EJERCICIOS 2 Y 3.

?? 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 INDESCRIFRABLE Y EXPRESIONES MALSONANTES.
 ?? UTILIZAR PARA EL DISEÑO LOS CRITERIOS DE INGENIERÍA (MÁXIMA EFICIENCIA, MÍNIMO COSTO).
 ?? UTILIZAR COMENTARIOS EN LA ELABORACIÓN DEL CÓDIGO SW.

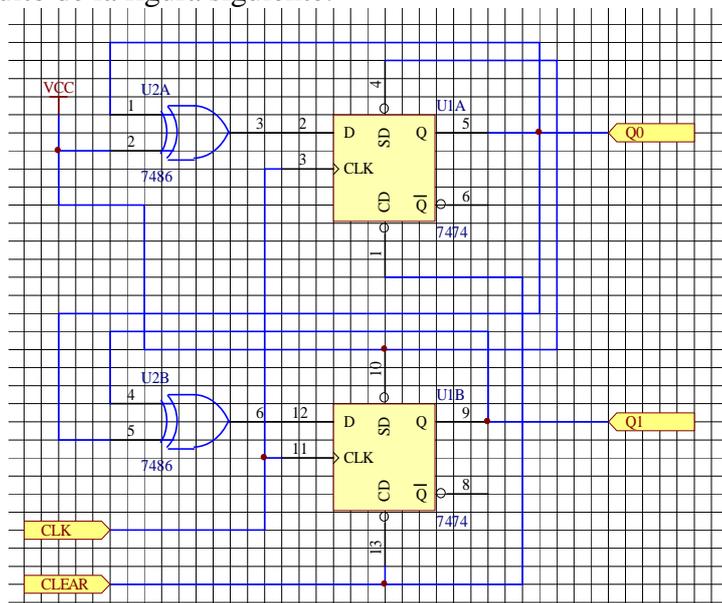
P1. Dado el diagrama de flujo siguiente:



Se pide:

- a) Tabla de Estados (1 punto).
- b) Reducción del diagrama mediante el método de la tabla de implicación de estados (3 puntos).
- c) Diagrama de estados resultante (1 punto).

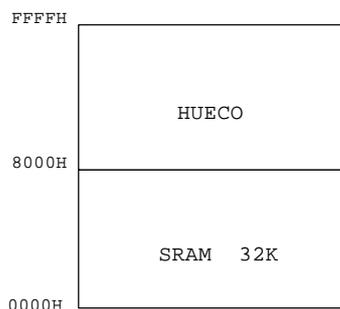
P2. Dado el circuito de la figura siguiente:



Se pide:

- Ecuaciones de excitación de los biestables (1 punto).
- Cronograma del sistema secuencial (2 puntos).
- Diagrama de flujo del sistema secuencial (1 punto).
- Teniendo en cuenta que: T_p (puertas lógicas)=15 ns, T_p (flip-flop)=10ns, T_{set-up} (flip-flop)=2 ns, T_{hold} (flip-flop)=3ns, determinar cuál será la frecuencia máxima de la señal de reloj a la que puede funcionar el circuito anterior (1 punto).

P3. Un sistema basado en el microcontrolador 8031 funcionando a 12MHz dispone de RAM y EPROM externa así como de cuatro puertos de salida realizados mediante 4 74LS374. El sistema dispone un área XDATA de 32Kb de RAM con un mapa de memoria como el especificado en el dibujo siguiente.



- Diseñar la circuitería lógica necesaria para demultiplexar el bus de direcciones y controlar la memoria RAM del sistema (señales ALE, /RD y /WR del 8051 y /CS1, CS2, /OE de la RAM) (2 puntos).
- Especificar la circuitería lógica necesaria para generar las señales de control de los 4 dispositivos periféricos (decodificador de direcciones) situados en cualquier dirección(es) del sistema a partir de la dirección 8000H de forma que no interactúe con la zona ocupada por la SRAM empleando el mínimo número de componentes y especificar el interfaz hardware/software, o sea las direcciones de memoria ocupadas por los registros de solo escritura (3 puntos).

P4. 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.

- Dibuja el organigrama del proceso de lectura anterior (1 punto).
- 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).
- Escribe un programa principal que llame a esta función y muestre en pantalla empleando "printf" el número de letras que se ha leído en este formato:
letra "c": 10.
letra "a": 3.
Solo se mostrarán las letras que existan (2 puntos).