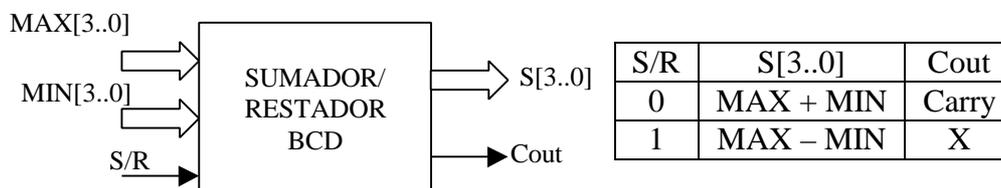


APELLIDOS : _____
 NOMBRE: _____ D.N.I.: _____

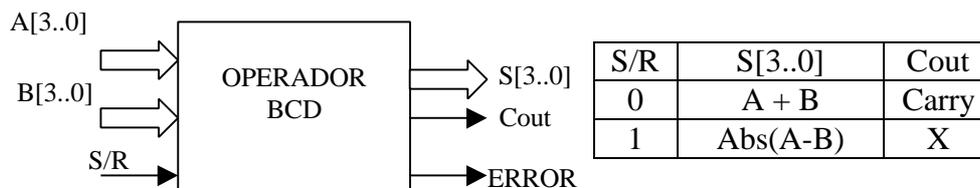
- ?? LEER ATENTAMENTE EL ENUNCIADO.
- ?? RAZONAR CADA UNA DE LAS DECISIONES ADOPTADAS TANTO EN EL HW COMO EN EL SW.
- ?? CUIDAR LA PRESENTACIÓN EVITANDO ESQUEMAS ININTELIGIBLES Y LETRA INDESCIFRABLE.
- ?? UTILIZAR PARA EL DISEÑO LOS CRITERIOS DE INGENIERÍA (MÁXIMA EFICIENCIA, MÍNIMO COSTO).
- ?? UTILIZAR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

P1. (3 puntos)

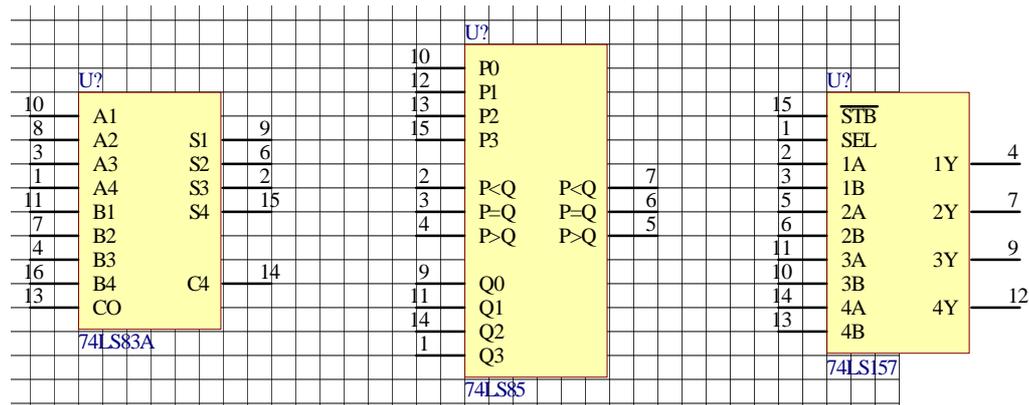
- a) Diseñar un bloque combinacional que reciba como entrada dos palabras de 4 bits representando dos dígitos BCD y genere una señal de ERROR que deberá activarse (valor alto) cuando alguna de dichas palabras de entrada no corresponda a un valor válido BCD. Usar las puertas lógicas necesarias.
- b) Diseñar un bloque sumador/restador como el de la figura en el que se introduce por las entradas etiquetadas como MAX[3..0] y MIN[3..0] el máximo y mínimo respectivamente de dos dígitos BCD. La señal S/R determina la operación a realizar de acuerdo con la siguiente tabla. Usar el bloque sumador 74LS83 y la lógica adicional necesaria.



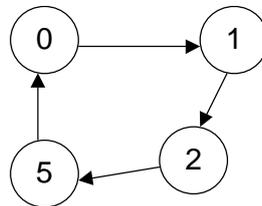
- c) El modulo de la figura representa un circuito combinacional que realiza las operaciones de suma (A+B) y valor absoluto de la diferencia (abs(A-B)) con operandos BCD. La señal de ERROR es activa (nivel alto) cuando A[3..0] o B[3..0] no corresponden a dígitos BCD. Las señales S[3..0] y Cout se describen en la tabla adjunta.



Diseñar dicho módulo utilizando exclusivamente los bloques diseñados en los apartados a) y b) y las unidades necesarias de los bloques comparadores 74LS85 y multiplexores 74LS157 (sin lógica adicional).

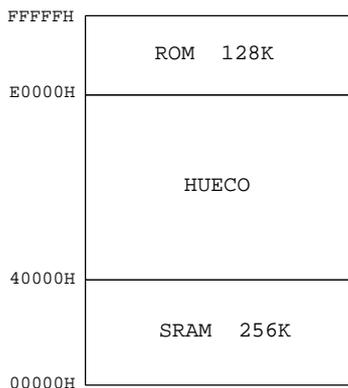


P2. (3 puntos) Realiza un contador síncrono que produzca la secuencia de la figura codificada en binario. Utilizar biestables tipo T y la lógica adicional necesaria. Si el sistema pasa por error por un estado no válido, el próximo estado debe ser válido.



P3. (4 puntos) A un sistema basado en el microprocesador 8086 funcionando a 8MHz en modo mínimo se le desea conectar una interfaz para controlar un periférico de E/S 8255 en la dirección de E/S 0AF0H para controlar ocho LEDs y ocho pulsadores. El periférico 8255 y las memorias se conectan a un bus del sistema ya demultiplexado, donde se dispone de las señales M/IO, /RD, /WR, A[19..0], D[15..0].

El sistema dispone 256Kb de RAM y 128Kb de ROM de un mapa de memoria como el especificado en el dibujo siguiente.



Este mapa se realiza mediante memorias SRAM estándar de 128Kx8 y mediante EPROM de 64Kx8.

- Diseñar la circuitería lógica necesaria para generar todas las señales de control de las memorias RAM.
- Diseñar la circuitería lógica necesaria para generar las señales de control de las memorias ROM.
- Diseñar la circuitería lógica de control del 8255, que genera sus seis señales de control. Conectar los LEDs al puerto A (PA[7..0]) y el teclado al puerto B (PB[7..0]) y dibujar su esquemático. Especificar el interfaz hardware/software, o sea, en que dirección del espacio de E/S se ubica cada registro de lectura y/o escritura del Interfaz de Periféricos.
- Diseñar un programa en lenguaje ensamblador que lea el teclado y encienda el LED correspondiente a la tecla pulsada. El programa se ejecutará continuamente.