

## EJERCICIOS. (Tema 5).

- 1) Diseña un circuito que sume la unidad a un número de  $N$  bits.
- 2) Se desea diseñar un circuito que realice la resta de dos números positivos ( $A$  y  $B$ ) de 4 bits. Para ello, es necesario convertir el número  $B$  positivo en negativo utilizando la representación en complemento a dos. Otra restricción para el circuito será que únicamente se dispone de registros de 4 bits para las operaciones internas. Finalmente, la salida del sistema deberá ser “corregida” en caso de que el resultado sea negativo, es decir, la salida del circuito no debe ser el resultado en complemento a dos, sino que debe darse en binario natural y mediante un led indicar que el resultado es negativo. Podría decirse que la representación de la salida del circuito será en formato signo-magnitud, donde el signo lo indicará el led.
- 3) Se desea diseñar un circuito que realice la resta o la suma de dos números de 4 bits que puedan ser positivos o negativos. Por ello, el circuito dispondrá de una entrada adicional que permita al usuario seleccionar la operación a realizar (suma o resta). Los números negativos deben representarse en complemento a dos y la salida del sistema deberá ser “corregida” tal y como se ha explicado en el ejercicio anterior.
- 4) Diseña un circuito que multiplique un número de  $n$  bits por cinco.
- 5) Diseña un multiplicador de  $2\text{bits} \times 2\text{bits}$ . Utilizarlo como módulo básico para hacer un multiplicador de  $6\text{bits} \times 6\text{bits}$ .
- 6) Diseña un circuito que realice la suma de dos números expresados en BCD natural. La salida del circuito también será en BCD natural.
- 7) En binario natural el complemento a 1 se genera simplemente invirtiendo todos los bits. Lo equivalente en base 10 es el complemento a 9 y se genera obteniendo la diferencia a 9 de cada dígito decimal. Así, el complemento a 9 del número 164 será el 835. Diseñar un circuito *complementador a 9* de números expresados en código BCD. Realizar el circuito mediante el mínimo número de puertas lógicas.
- 8) Diseña un circuito que realice la suma de dos números expresados en BCD exceso tres. La salida del circuito también será en BCD exceso tres.
- 9) Utilizando el decodificador 74154, implementa un circuito para decodificar un número de 5 bits. Determinar la salida que se activa al introducir el código binario de entrada 10110.
- 10) Diseña un decodificador de 2 a 4 que disponga de una entrada de activación  $E$ . Todas las entradas así como las salidas serán activas a nivel alto.  
Utilizando el decodificador diseñado como unidad básica, implementa un decodificador de 4 a 16.
- 11) Utiliza multiplexores 74150 y cualquier otra lógica necesaria para multiplexar 32 líneas de datos en una única línea de salida.
- 12) Utiliza el C.I. 74151 para implementar la función:  $f = \Sigma m(0, 2, 3, 5)$
- 13) Utilizando un multiplexor de 4 a 1, implementa la función  $f(a, b, c) = ab + \bar{b}c$
- 14) Utilizar el C.I. 74151 para implementar la función:  $f = \Sigma m(0, 1, 2, 3, 4, 9, 13, 14)$

15) Diseña una ALU que realice las siguientes operaciones sobre dos números de 4 bits. En caso de que la salida debe mostrar el resultado aritmético, éste deberá aparecer “corregido” tal y como se ha explicado en el ejercicio 2.

<u>Aritméticas</u>	<u>Lógicas</u>
Sumar sin acarreo	AND
Sumar con acarreo	OR
Decrementar A en una unidad	NOT A
Transferir A	$A \oplus B$
Incrementar A en una unidad	
Restar en complemento a 1	
Restar en complemento a 2.	

16) Diseña una ALU que realice las siguientes operaciones sobre dos números de 4 bits. En caso de que la salida debe mostrar el resultado aritmético, éste deberá aparecer “corregido” tal y como se ha explicado en el ejercicio 2.

<u>Aritméticas</u>	<u>Lógicas</u>
Sumar sin acarreo	AND
Sumar con acarreo	OR
Restar sin acarreo	NOT A
Restar con acarreo	NOT B
Incrementar A en una unidad	$A \oplus B$
Decrementar A en una unidad	NAND
Transferir A	NOR
Obtener el complemento a 1 de B	
Obtener el complemento a 2 de B	
Multiplicar A por 2	
Poner la salida a 0000	
Poner la salida a 1111	

17) Indica qué minterminos conforman la función F que aparece en la Figura 1. Simplifica la función y exprésala como suma de productos y como producto de sumas. Finalmente, implementala función con el C.I. 17454 y el número mínimo de puertas NAND.

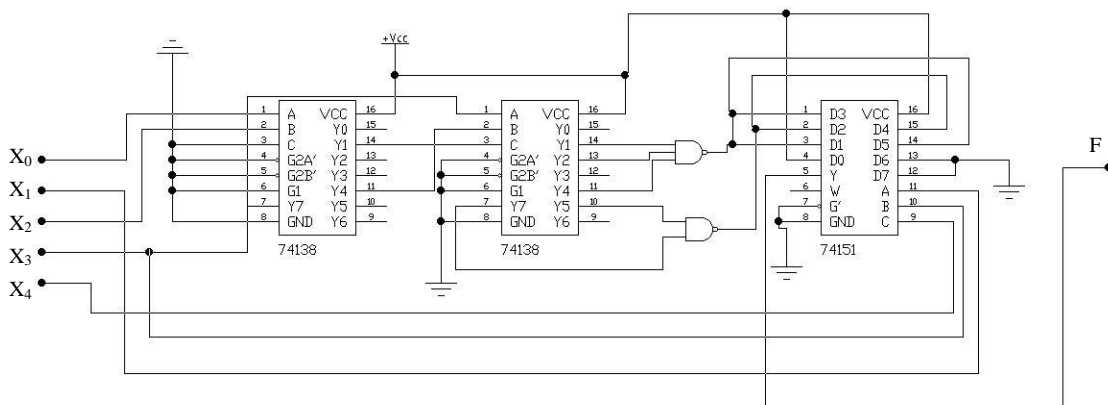


Figura 1

18) Un sistema digital tiene como entradas tres grupos de seis bits cada uno, uno de ellos será la referencia R para los otros dos grupos, A y B. Para cada grupo, los dos primeros bits son de control y los restantes un número decimal representado en BCDnat. Realizar el circuito que dé como salida un número en BCDnat, cumpliendo con las siguientes especificaciones:

- a) Si los dos primeros bits ( $R_1R_0$ ) de R son 00, la salida debe ser cero.
- b) Si los dos primeros bits ( $R_1R_0$ ) de R son distintos de 00, la salida será el *mayor* de los números decimales contenidos en A o B, siempre que sus bits de control ( $A_1A_0$ ,  $B_1B_0$ ) sean 00 o 10 y el decimal correspondiente sea *menor* que el que contiene R. De no ser así la salida será cero.

Utilizar los circuitos MSIs necesarios y el mínimo número de puertas lógicas necesarias. No es necesario dar el número de los MSI, basta con decir el tipo a utilizar. Por supuesto, debe de existir en el mercado.

19) Un expendedor de golosinas recibe por un lado, codificado en binario natural, el dinero introducido y por otro lado recibe proveniente de cuatro interruptores G, Ch, Re, Pi (Gominolas, Chicle, Regaliz, Pipas) el producto solicitado. Cada producto anterior cuesta 2, 3, 5 y 12 céntimos de euro respectivamente, y las monedas a introducir serán de 5 céntimos, de 1 céntimo y de 2 céntimos de euro. Se supone que el usuario es una persona inteligente y si quiere comprar una golosina que cuesta 2 céntimos como mucho introducirá una moneda de 5 céntimos, ya que introducir 2 monedas de 5 céntimos no tendría sentido. Sí puede ocurrir que elija un producto y no introduzca todo el dinero necesario. Implementar el circuito que visualiza en *displays* de 7-segmentos cuánto dinero falta o sobra y en dos diodos *led* si se debe entregar el producto o si el usuario debe introducir más dinero.

- a) Realizar un diagrama de bloques del circuito, indicando claramente las entradas y salidas de cada bloque (qué representan, número de bits razonado, número máximo y mínimo a obtener, registro a utilizar razonado...).
- b) Realizar el diseño óptimo de cada bloque del apartado anterior, utilizando los circuitos integrados que se describen en la hoja anexa y el mínimo número de puertas lógicas. Determinar el tipo de *display* a utilizar y realiza las conexiones adecuadas para que cuando el resultado sea cero, se visualice un único cero y cuando el resultado contenga un único dígito, el display correspondiente al dígito de mayor peso no indique un cero.

**Nota 1:** Todas las entradas utilizables de los circuitos integrados deben estar conectadas correctamente.

**Nota 2:** No es necesario que el orden de las patillas de los circuitos integrados sea el que aparece en la hoja de características, a la hora de dibujarlos.

20) Se desea diseñar un sistema de control de acceso a un recinto privado. Para ello, los visitantes deben conseguir que se abran tres puertas que funcionan de la siguiente manera:

- 1) Para abrir la puerta P1, hay que introducir un dígito BCD mayor o igual que 4 y que sea par.
- 2) Para abrir la puerta P2, hay que haber abierto P1 y hay que introducir un número que esté entre 2 unidades por arriba y 2 unidades por debajo de la mitad del número introducido por la puerta P1.
- 3) La puerta P3 se abrirá si se han abierto las anteriores y además hay sitio libre en el recinto ya que no se ha llegado al aforo máximo.

Las entradas del sistema a diseñar serán:

- 1) Teclado numérico, es decir, 10 interruptores donde cada uno de ellos representa un número de 0 a 9, para introducir el número para la puerta P1.
- 2) Teclado numérico, es decir, 10 interruptores donde cada uno de ellos representa un número de 0 a 9, para introducir el número para la puerta P2.
- 3) Entradas para que el portero indique, cada día, cuál será el aforo máximo. Se introducirá en binario y como mucho podrá ser 25.
- 4) Sensores, S1 y S2, que indican mediante un pulso si ha entrado o salido una persona del recinto. El sensor S1 (entrada) está detrás de la puerta P3 por lo tanto sólo se activará pasada esta puerta. El sensor S2 (salida), está en la puerta de salida del recinto.
- 5) Interruptor para que el visitante pueda seleccionar entre ver el número de plazas libre o el aforo máximo del día.

Salidas del sistema

- 6) Abrir puerta P1
- 7) Abrir puerta P2
- 8) Abrir puerta P3
- 9) Dos *Displays* para visualizar el aforo máximo o el número de plazas libres en los mismos *displays*. Configura el circuito para conseguir la supresión del cero del dígito de mayor peso.

Una vez superada la puerta P3, ésta y las anteriores (P1 y P2) se cierran mediante un sistema externo automático. Igualmente, este sistema desactiva las entradas de los teclados dejándolas listas para el siguiente cliente.

**Realizar un diagrama de bloques del circuito, indicando claramente las entradas y salidas de cada bloque (qué representan, número de bits razonado, número máximo y mínimo a obtener, registro a utilizar razonado ...).**

**Realizar el diseño óptimo de cada bloque del apartado anterior.**

**Determinar el tipo de display a utilizar.**

**Nota 1:** Los circuitos que se DEBEN utilizarse son: 74183, 74147, Contador (ver en nota 7 su funcionamiento, 7483)

**Nota 2:** Todas las patillas de los integrados deben estar correctamente conectadas. No es necesario ponerlas en el mismo orden que en el diagrama.

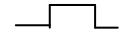
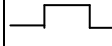
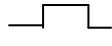
**Nota 3:** Los números negativos en complemento a dos.

**Nota 4:** Los bloque que haya que desarrollar funciones lógicas a partir de tablas de verdad, basta con indicar los mintérminos o los maxtérminos que la componen.

**Nota 5:** Para dividir un número en binario entre dos, basta con desplazar los bits hacia las derecha en una posición.

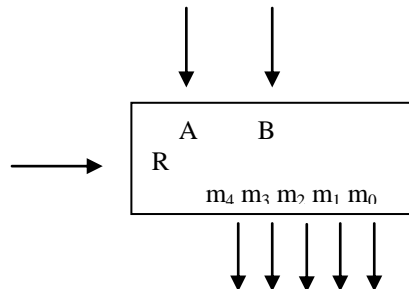
**Nota 6:** Únicamente se permiten utilizar **dos** circuitos integrados Comparadores (7485)

**Nota 7: Contador +1 o -1**

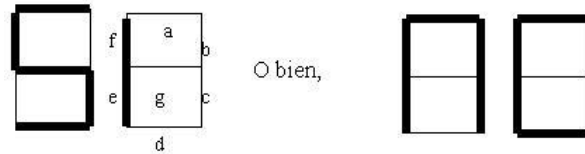
A	B	R	Operación
0	0		Salidas a cero
0		0	Aumenta en 1 la salida
	0	0	Disminuye en 1 la salida

 Representa un pulso

La salida sólo representa números positivos.



21) Sean 6 provincias (A, B, C, D, E, F) que votan diversas propuestas en el parlamento. El resultado de la votación será SI o NO visualizados en dos *displays* de 7 segmentos. En caso de empate, decide la provincia A. Además, la provincia anfitrión tiene un interruptor de trampa (T) para dar la vuelta al resultado, de modo que se represente el NO a pesar de haberse aprobado la propuesta y viceversa. Finalmente, dispone de otra entrada (N/P) para disponer que se represente el número de votos afirmativos en vez de las palabras SI/NO. De ese modo, si  $N/P=1$ , la visualización en el *display* será, opcionalmente:



Si  $N/P=0$ , en los *displays* debe representarse el número de votos afirmativos. Por motivos de imagen pública, en caso de hacer trampa ( $T=1$ ), se visualiza el SI o el NO, independientemente del valor que tome  $N/P$ .

- Realizar un diagrama de bloques del circuito, indicando claramente las entradas y salidas de cada bloque (qué representan, número de bits razonado, número máximo y mínimo a obtener, registro a utilizar razonado ...).
- Realizar el diseño óptimo de cada bloque del apartado anterior, utilizando los circuitos integrados que se describen en la hoja anexa y el mínimo número de puertas lógicas. Determinar el tipo de *display* a utilizar.

**Nota 1:** Todas las entradas utilizables de los circuitos integrados deben estar conectadas correctamente.

**Nota 2:** No es necesario que el orden de las patillas de los circuitos integrados sea el que aparece en la hoja de características, a la hora de dibujarlos.

**Nota 3:** Los circuitos que haya que diseñar mediante tablas de verdad y mapas de *Karnaugh*, no es necesario dibujarlos ni realizar la simplificación. Basta con expresar la tabla de verdad con claridad y la expresión canónica de cada función.

**Nota 4:** Obsérvese que no se permite utilizar comparadores, por lo que se utilizará la operación *resta* para determinar si un número es mayor que otro. Para ello, se utilizará la codificación en complemento a dos.

22) Especificaciones del sistema a diseñar:

- Se dispone de dos teclados decimales (teclas de 0 a 9) con teclas como interruptores no pulsadores. Cada tecla representa un dígito decimal, cuyos estados lógicos significan que se ha pulsado un dígito decimal o que no se ha pulsado. En caso de pulsar más de una tecla a la vez únicamente tendrá valor el número decimal de mayor peso.
- También se dispone de dos entradas o una sola (como se crea más conveniente para el diseño) de operación que representan una operación aritmética: S será suma y la M multiplicación. En caso de realizar el diseño con dos entradas nunca se podrán pulsar las dos simultáneamente.
- Una entrada de Reset pone los operandos a 0.
- El resultado del circuito debe representarse en dos displays de 7 segmentos.

El funcionamiento será:

- Se introducen los operandos a través del teclado.
- Se introduce la operación a realizar: S o M.
- El resultado se presenta en los displays.

Se pide realizar el diagrama de bloques especificando tipos de CIs a utilizar (no es necesario indicar el número en este apartado) y funcionalidad de los circuitos que haya que diseñar. Todas las variables de entrada y salida de cada circuito y las conexiones entre los distintos bloques deben estar claramente especificadas.

CIs utilizables cuyas hojas de características se adjuntan: 74147, 74148, 7400, 74185A, 7483, 7447, 7448, 74157, Semisumadores, Sumadores completos, displays de 7 segmentos de ánodo común, displays de 7 segmentos de cátodo común.

23)

a) Diseñar una Unidad Aritmético Lógica que realiza las siguientes operaciones.

Aritméticas	Lógicas
A - B (En complemento a 1)	$\overline{B}$
A + B	$A + \overline{B}$
A + 1	$A \oplus \overline{B}$
A * 2	A.B
A + 0	$A \oplus B$
	A+B

Realiza el diseño siguiendo el esquema explicado en clase y teniendo en cuenta:

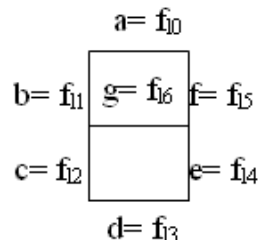
- ✓ La operación aritmética A - B debe realizarse en C1.
- ✓ Los números A y B que introduce el usuario serán números codificados en BCD Natural.
- ✓ La ALU tendrá como salidas:
  - *led-de-operación* que indicará si la operación que se está realizando es lógica o aritmética.
  - *led-de-signo* que indicará si el resultado de la operación aritmética es positivo o negativo.
  - bits necesarios para que en caso de haber realizado una operación aritmética, se lea en ellos el módulo del resultado en binario natural, ya que el signo nos lo muestra el *led-de-signo*.

**Nota 1:** Los circuitos a utilizar pueden seleccionarse en el apartado correspondiente de la hoja anexa.

**Nota 2:** Todas las patillas de los integrados deben estar correctamente conectadas. No es necesario ponerlas en el mismo orden que en el diagrama.

b) Añadir una etapa para poder visualizar los resultados de la siguiente manera:

- El resultado de la operación aritmética, debe visualizarse a través de los *displays* de 7-segmentos.
- El resultado de la operación lógica se visualizará en el *display* de menor peso asignando a cada *bit* un segmento de la siguiente manera:



Donde  $f_i$  son los resultados de la operación lógica.

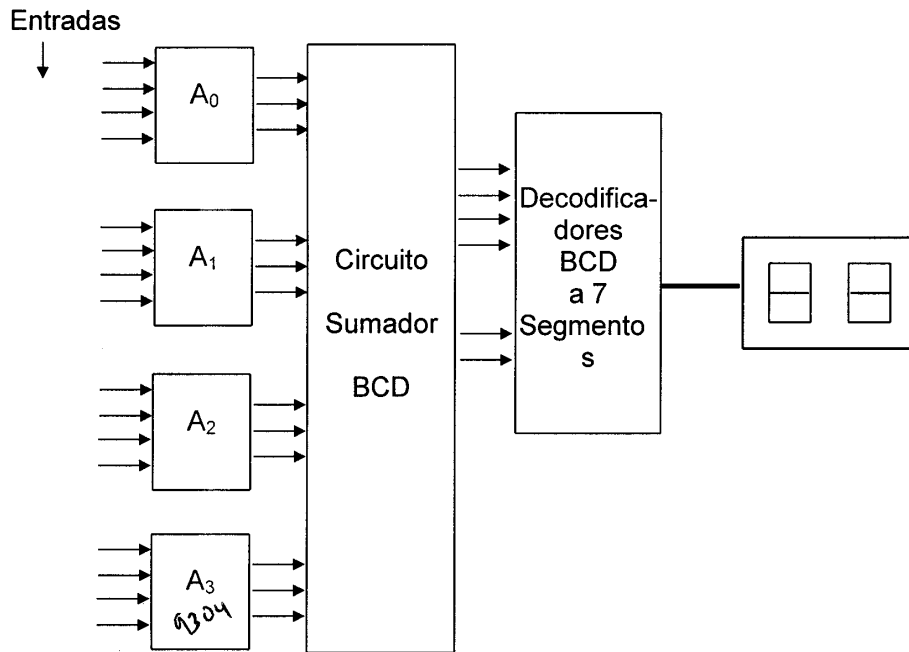
Indica el tipo de *display* a utilizar y configura el circuito para conseguir la supresión del cero del dígito de mayor peso tanto para el resultado de las operaciones aritméticas como para el de las lógicas.

**Nota 1:** Los circuitos a utilizar pueden seleccionarse en el apartado correspondiente de la hoja anexa.

**Nota 2:** Todas las patillas de los integrados deben estar correctamente conectadas. No es necesario ponerlas en el mismo orden que en el diagrama.

24) Para una asamblea, donde participan veinte (20) personas, se desea hacer una máquina que sume los votos afirmativos y muestre la cantidad en un indicador numérico (display 7 segmentos). Para ello cada asambleísta, tiene acceso a un interruptor, por medio del cual hace su voto, de manera que un "1" indique "sí" y un "0" indique "no". Los interruptores se deberán agrupar de cinco en cinco, para obtener cuatro sumas parciales. Los resultados de cada grupo, deberán ir a un circuito desarrollado con bloques lógicos funcionales y compuertas, y obtener el resultado de la votación en BCD Natural y 7 segmentos (dos dígitos).

El diagrama de bloques propuesto, se presenta a continuación:



Para la realización de los bloques que aparecen en el diagrama podrán utilizarse únicamente los siguientes integrados: Sumadores Completos, 7483 (sumador de dos números de 4 bits), 7447 (decodificador BCD-7segmentos), display de 7 segmentos, y las puertas lógicas necesarias. Las hojas de características se adjuntan.

Todos los pines de los integrados deben estar correctamente conectados.

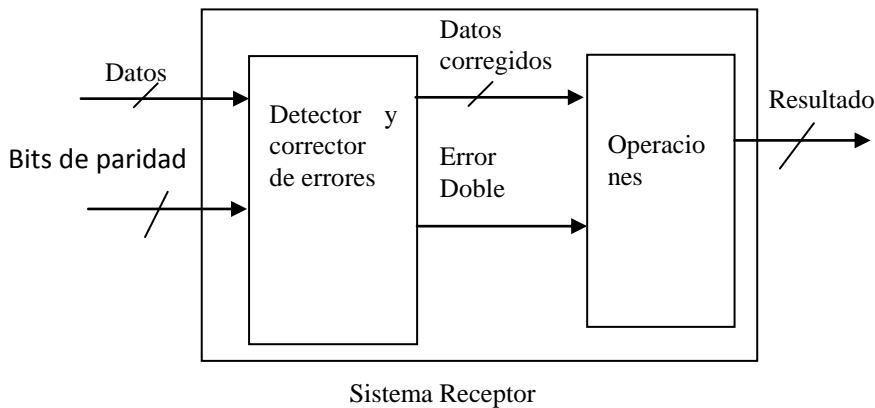
Debe especificarse el tipo de display a utilizar. Si la suma de los votos consta únicamente de un dígito, el display de mayor peso deberá permanecer desactivados.



25) Un sistema de transmisión envía palabras de cuatro bits de datos a un centro de control, utilizando códigos *Hamming* con paridad impar. En dicho centro de control se chequea la información llegada. Si ha llegado correctamente, se realizan ciertas operaciones con la palabra de 4 bits. Si se detecta que ha ocurrido un error en la transmisión, se corrige. Si no es posible su corrección, no se realiza ninguna operación con la palabra recibida. El sistema puede corregir un bit y detectar dos errores.

Las operaciones a realizar son: si el dato enviado está entre 7 y 15 (ambos inclusive) se le debe restar 4 utilizando complemento a dos. Si no, se le suma cinco.

Se pide diseñar el sistema receptor que presenta la siguiente estructura:



- Realizar un diagrama de bloques de cada uno de los bloques principales del sistema receptor, indicando claramente las entradas y salidas de cada bloque (qué representan, número de bits, ...).
- Realizar el diseño óptimo de cada bloque del apartado anterior, utilizando los circuitos integrados 7404, 7408, 7432, 7485, 74139, 7482, 74157, 7447, y el mínimo número de puertas lógicas.
- Añadir una etapa final al diseño que permita visualizar el resultado en *displays* de 7 segmentos, con supresión del cero a la izquierda. Realizar el diseño, indicando y justificando el tipo de integrado a utilizar y el mínimo número de puertas lógicas.

**Nota 1:** Todas las entradas utilizables de los circuitos integrados deben estar conectadas correctamente.

**Nota 2:** No es necesario que el orden de las patillas de los circuitos integrados sea el que aparece en la hoja de características, a la hora de dibujarlos.

26) Sea una nevera para restaurantes con dos compartimentos: uno para el vino y otro para la carne. El compartimiento del vino debe mantenerse entre  $14-17\text{ }^{\circ}\text{C}$  y el de la carne entre  $-3$  y  $4\text{ }^{\circ}\text{C}$ . Para detectar estas temperaturas, se dispone de sendos sensores que generan en su salida la temperatura en binario natural y en complemento a dos para los casos de temperaturas bajo cero. Los rangos para estos sensores son de  $-20$  y  $30\text{ }^{\circ}\text{C}$ . Se desea diseñar un sistema combinacional que detecte si alguno de los compartimentos está fuera de rango. Asimismo, la nevera dispondrá de una entrada de selección que permita ver la magnitud de los grados de uno u otro compartimiento en un único sistema de *displays*. En caso de estar fuera de rango la representación en los *displays* debe ser 00. Igualmente, en caso de que la temperatura sea negativa, deberá indicarse mediante un *led* y observar la magnitud en los *displays*.

Realizar un diagrama de bloques del circuito, indicando claramente las entradas y salidas de cada bloque (qué representan, número de bits razonado, número máximo y mínimo a obtener, registro a utilizar razonado...).

27) La entrada a un circuito la constituyen tres grupos de 3 bits  $A$ ,  $B$ ,  $C$ . Asimismo, la salida está formada por dos grupos de 3 bits  $M$  y  $m$ , y  $R$ . La salida  $M$  representa el mayor de los tres grupos de bits de entrada y la salida  $m$  el menor de ellos. Por su parte, la salida  $R$  es el resultado de una operación entre el número obtenido en  $M$  y en  $m$ . Dicha operación depende del valor de  $C$ . Si el número que representa  $C$  está entre el 2 y el 5 (ambos inclusive), entonces se realiza la resta en complemento a dos ( $M-m$ ). En caso contrario, se realiza la suma ( $M+m$ ).

- Realizar un diagrama de bloques del circuito, indicando claramente las entradas y salidas de cada bloque (qué representan, número de bits, ...).
- Realizar el diseño óptimo de cada bloque del apartado anterior, utilizando los circuitos integrados 7493, 7485, 74157, 7447 y el mínimo número de puertas lógicas.
- Añadir una etapa final al diseño que permita visualizar el resultado ( $R$ ) en *displays* de 7 segmentos, con supresión del cero a la izquierda. Realizar el diseño, indicando y justificando el tipo de integrado a utilizar y el mínimo número de puertas lógicas.

**Nota 1:** Todas las entradas utilizables de los circuitos integrados deben estar conectadas correctamente.

**Nota 2:** No es necesario que el orden de las patillas de los circuitos integrados sea el que aparece en la hoja de características, a la hora de dibujarlos.