

Problema 1 (2 puntos)

- Si $A=202$ está expresado en octal, convertirlo a binario natural, hexadecimal, decimal y BCD.
- Si $A=1100110$, determine cuál es su valor decimal en los siguientes casos:
 - Si se interpreta como un valor binario natural sin signo.
 - Si se interpreta como un valor en signo-magnitud.
 - Si se interpreta como un valor con signo en complemento a 1.
 - Si se interpreta como un valor con signo en complemento a 2.
- Si $A=0110110$ y $B=1101101$ son números con signo expresados en complemento a dos, realice las operaciones $A+B$, $A-B$, $-A+B$ y $-A-B$, deduzca de forma justificada el valor decimal del resultado y si se produce desbordamiento en cualquiera de los casos

Solución:

1. (0,5 puntos) $A=202_8 = \frac{2}{10} \frac{0}{000} \frac{2}{010}_2 = 82_{16} = 16 \times 8 + 2 = 130_{10}$
 $= \frac{0001}{1} \frac{0011}{3} \frac{0000}{0} \text{BCD}$

2. (0,5 puntos)

a. $A = 1100110_2 = 66_{16} = 102_{10}$

b. $A = 1100110_{SM} = -\left(\frac{100110}{2} \frac{6}{6}\right)_2 = -(16 \times 2 + 6)_{10} = -38_{10}$

c. $A = 1100110_{CA1} = -\left(\frac{011001}{1} \frac{9}{9}\right)_2 = -(16 + 9) = -25_{10}$

d. $A = 1100110_{CA2} = -(CA_2(A)) = -\left(\frac{0011010}{1} \frac{10}{10}\right)_2 = -26_{10}$

3. (1 punto)

(*) $A = 0110110 (54_{10})$
 $+ B = 1101101 (-19_{10})$
 $A+B = 10100011 = 35_{10}$, no hay overflow

(*) $A-B = A + CA_2(B)$; $CA_2(B) = 0010011 = 19_{10}$
 $A = 0110110 (54_{10})$
 $+ CA_2(B) = 0010011 (+19_{10})$
 $A-B = 01001001 (+73_{10})$ Hay overflow pues la suma de 2 nros positivos da un nro negativo. Hace falta 1 bit más para el resultado

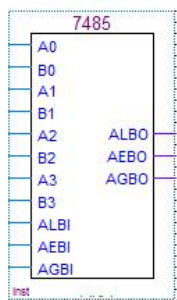
(*) $-A-B = CA_2(A) + CA_2(B)$; $CA_2(A) = 1001010 = -54_{10}$
 $-A = 1001010$
 $+ -B = 0010011 = -19_{10}$
 $-A-B = 1011101 = -(0100011)_2 = -35_{10}$, no hay overflow

También:
 $-A-B = -(A+B)$
 $-A+B = -(A-B)$

(*) $-A+B = CA_2(A) + B = (1001010) + (1101101) = 0110111$
 Hay overflow, se necesita un bit más $\Rightarrow -A+B = 10110111 = -(01001001)_2 = -73_{10}$

Problema 2 (2,5 puntos)

Dado el circuito de la figura y su tabla de verdad:



Comparing Inputs				Cascading Inputs			Outputs		
A3, B3	A2, B2	A1, B1	A0, B0	AGBI	ALBI	AEBI	AGBO	ALBO	AEBO
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

H = HIGH Level, L = LOW Level, X = Don't Care

Se pide:

- Describa en lenguaje natural la función que realiza dicho circuito. Razone su respuesta.
- Realice el programa VHDL mediante un proceso.

Solución:

- (0,5 puntos) El circuito es un comparador de 4 bits, que permite determinar si dos números de 4 bits codificados en binario natural (sin signo) son iguales, o si uno es mayor que el otro. El 1º número es A, formado por las entradas A3...A0, siendo A3 el bit más significativo y A0 el menos, el otro nº es B formado por B3...B0. Si observamos la tabla de verdad, la salida AGBO indica A > B, la salida ALBO indica A < B y la salida AEBO indica A = B. Analizando la tabla se deduce que las entradas AGBI, ALBI y AEBI significan igualmente A > B, A < B y A = B, y dado que solo influyen en las salidas cuando A = B, se utilizan para comparar números de más de 4 bits, para conectar las salidas de la comparación de una etapa que compara bits menos significativos a la etapa siguiente

NOMBRE:

b) (2 puntos)

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity COMP4 is
6      port (
7          A,B: in unsigned(3 downto 0);
8          AGBI,ALBI,AEBI: in std_logic;
9          AGBO,ALBO,AEBO: out std_logic
10     );
11 end COMP4;
12
13 architecture functional of COMP4 is
14     signal casc_in: std_logic_vector (2 downto 0);
15 begin
16     casc_in <= AGBI & ALBI & AEBI;
17
18     process (A, B, casc_in)
19     begin
20         AGBO <= '0';
21         ALBO <= '0';
22         AEBO <= '0';
23         if A > B then AGBO <= '1';
24         elsif A < B then ALBO <= '1';
25         else
26             case casc_in is
27                 when "100" =>
28                     AGBO <= '1';
29                 when "010" =>
30                     ALBO <= '1';
31                 when "001" | "011" | "101" | "111" =>
32                     AEBO <= '1';
33                 when "000" =>
34                     AGBO <= '1';
35                     ALBO <= '1';
36                 when others =>
37                     AGBO <= '-';
38                     ALBO <= '-';
39                     AEBO <= '-';
40             end case;
41         end if;
42     end process;
43
44 end functional;

```

Problema 3 (2,5 puntos)

Un coche autónomo posee, entre otros, cuatro sensores para evitar colisiones entre vehículos: sensor que indica si existe Movimiento (M), sensores que indican presencia de objetos LIDAR (L) y Cámara (C), por último, un sensor de proximidad Ultrasonidos (U), todos ellos activos a nivel alto.

NOTA: El orden de importancia de los bit sigue está codificación MLUC

El sistema anticolidión comprende dos elementos, el primero es activación del Sistema Pasivo de Anticolisión (SPA) en bloque (cinturón, airbag, ...) con el siguiente comportamiento nivel alto activado y nivel bajo desactivado, el segundo elemento es el control de intensidad de frenada (IF) que se codifica con dos bits:

Control de Intensidad de Frenada (IF)	Codificación
Ausencia de frenada (NF)	00
Frenada media (FM)	01
Frenada alta (FA)	11

El sistema se comporta de la siguiente manera: siempre que esta el vehículo en movimiento se activara el Sistema Pasivo de Anticolisión (SPA), en caso contrario estará desactivado el SPA. Siempre que al menos uno de los sensores de presencia detecte un objeto se activara la frenada media. En el caso que estén activos algún sensor de presencia y el sensor de proximidad se activaran la frenada alta. Cuando el coche este parado deberá estar activa frenada alta.

- A) Describir la entidad en VHDL del circuito, explicando la elección de cada tipo de dato (1,25 pts)

```
ENTITY Examen_18_19P1 IS
```

```
    PORT(
```

```
        MLUC :    IN          STD_LOGIC_VECTOR(3 DOWNTO 0);
        s_SP  :    OUT         STD_LOGIC;
        s_IF  :    OUT         STD_LOGIC_VECTOR (1 DOWNTO 0);
```

```
END Examen_18_19P1;
```


NOMBRE:

GRUPO:

B) Indicar la tabla de verdad (1,25 pts)

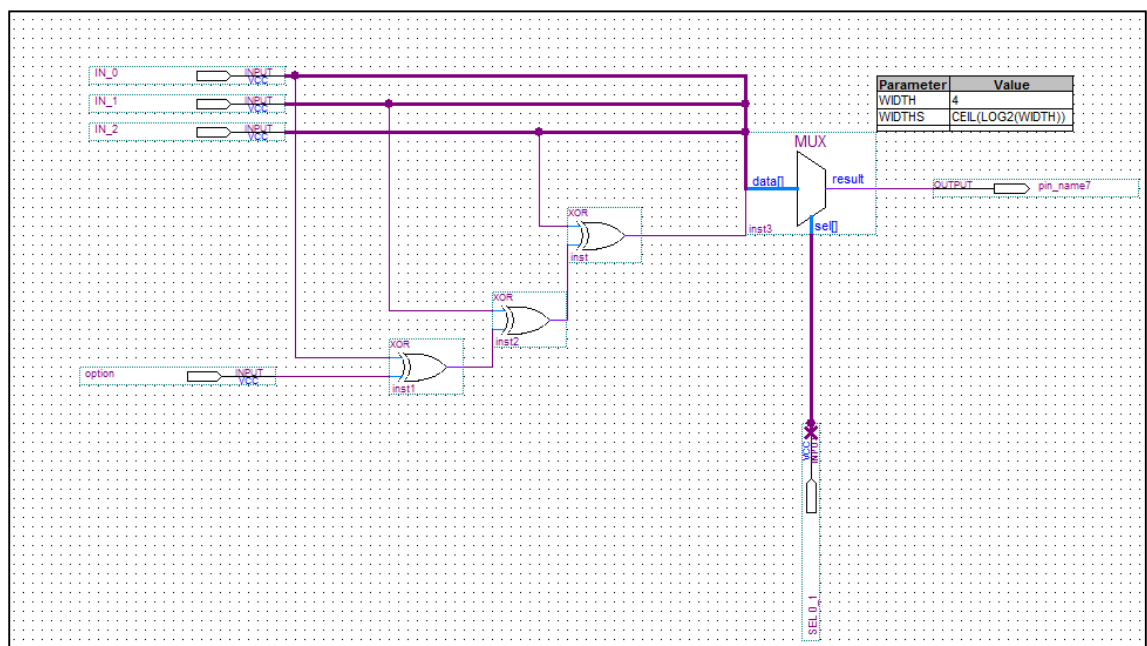
	Presencia de objeto					
M	L	C	U	SP	IF	
0	0	0	0	0	1	1
0	0	0	1	0	1	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	0	1	1
0	1	1	1	0	1	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	1	0	1
1	0	1	1	1	1	1
1	1	0	0	1	0	0
1	1	0	1	1	1	1
1	1	1	0	1	0	1
1	1	1	1	1	0	1
1	1	1	1	1	1	1

Problema 4 (3 puntos)

A) Obtener el circuito esquemático del siguiente código (1,25 pts)

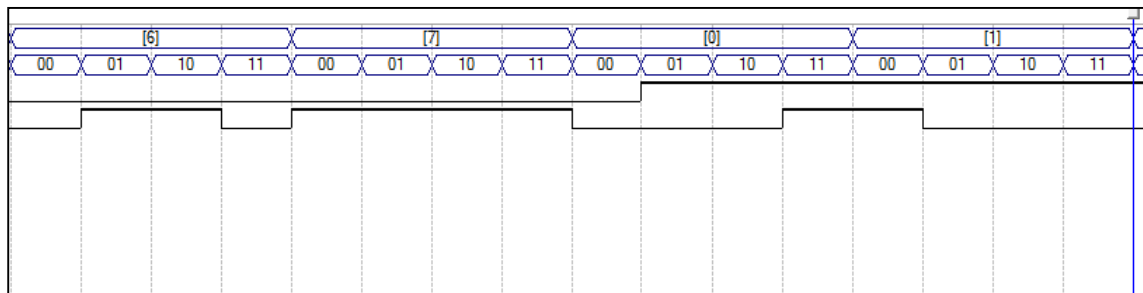
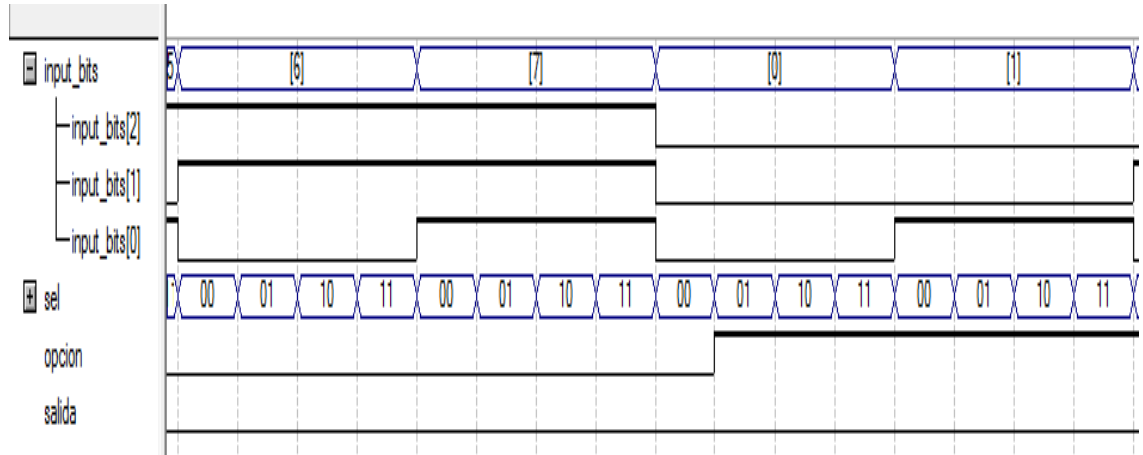
```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  ENTITY Examen_18_19P1 IS
5
6  PORT (
7
8      input_bits : IN      STD_LOGIC_VECTOR(2 DOWNTO 0);
9      opcion: IN STD_LOGIC;
10     sel: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
11     salida : OUT STD_LOGIC);
12 END Examen_18_19P1;
13
14 ARCHITECTURE logica OF Examen_18_19P1 IS
15     SIGNAL vector_intermedio : STD_LOGIC_VECTOR(3 DOWNTO 0);
16 BEGIN
17
18     PROCESS(input_bits, opcion, sel)
19     BEGIN
20
21         vector_intermedio(0) <= opcion;
22         vector_intermedio(1) <= vector_intermedio(0) XOR input_bits(0);
23         vector_intermedio(2) <= vector_intermedio(1) XOR input_bits(1); --BLOQUE OPCION--
24         vector_intermedio(3) <= vector_intermedio(2) XOR input_bits(2);
25
26         CASE sel IS
27             WHEN "00" => salida <= input_bits(0);
28             WHEN "01" => salida <= input_bits(1);
29             WHEN "10" => salida <= input_bits(2);
30             WHEN "11" => salida <= vector_intermedio(3);
31         END CASE ;
32
33     END PROCESS ;
34
35 END logica;
```



NOMBRE:

B) Dado los valores de entrada, complete la señal salida del cronograma (1,25 pts)



C) ¿Qué calculo/funcionalidad realiza el circuito, el BLOQUE OPCION sobre "input_bits", indicando para los dos valores de la entrada opción ('0' y '1')? (0,5 pts)

Bloque opción: Calcula la paridad de 'input' según el valor que tenga determina el tipo de paridad, para opción='0' es par en 1 y para opción='1' es impar en '1'