

NOMBRE Y APELLIDOS:

<b>Problema 1 (2 puntos)</b>	15 minutos
------------------------------	------------

1- **(50%)** Transforme 0xAB a

a. Decimal

b. Binario

c. Octal

d. BCD

2- **(50%)** Siendo  $A=+64_{10}$  y  $B=-120_{10}$

a. Obtenga A y B en Ca2 con 8 bits.

b. Realice las siguientes operaciones en Ca2 con 8 bits. Indique si hay acarreo y overflow. Justifique su respuesta.

i.  $A+B$

ii.  $A-B$

NOMBRE Y APELLIDOS:



Electrónica Digital. GICME GIT  
Primer Parcial. Curso 2021-2022  
27 de Octubre de 2021

<b>Problema 2 (4 puntos)</b>	35 minutos
------------------------------	------------

Para el siguiente circuito descrito en VHDL:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity prob2 is
.....(
    A,B:   IN std_logic_vector(1 downto 0);
    C:     IN std_logic;
    Z:     OUT std_logic_vector (1 downto 0);
    Y:     OUT std_logic_vector (1 downto 0);
    eo:    OUT std_logic;
end prob2;

architecture a_func of ..... is
    signal s1: std_logic_vector (..... downto 0);
    signal s2: unsigned (1 downto 0);
    signal s3: std_logic_vector(..... downto 0);

begin

    s1<=A xor B;
    s2<= unsigned(A)+ unsigned(B);
    s3<=A&B;

    p1: process(.....)
    begin
        if C='0' then
            Y<=s1;
        else
            Y<=std_logic_vector(s2);
        end if;
    end process;
  
```

```
p2:  process(.....)
      begin
          e0<='0';
          if C='1' then
              Z<="00";
          else
              case s3 is
                  when "0001" =>Z<="00"; e0<='1';
                  when "0010" =>Z<="01";
                  when "0100" =>Z<="10";
                  when "1000" =>Z<="11";
                  when others => Z<="00";
              end case;
          end if;
      end process;

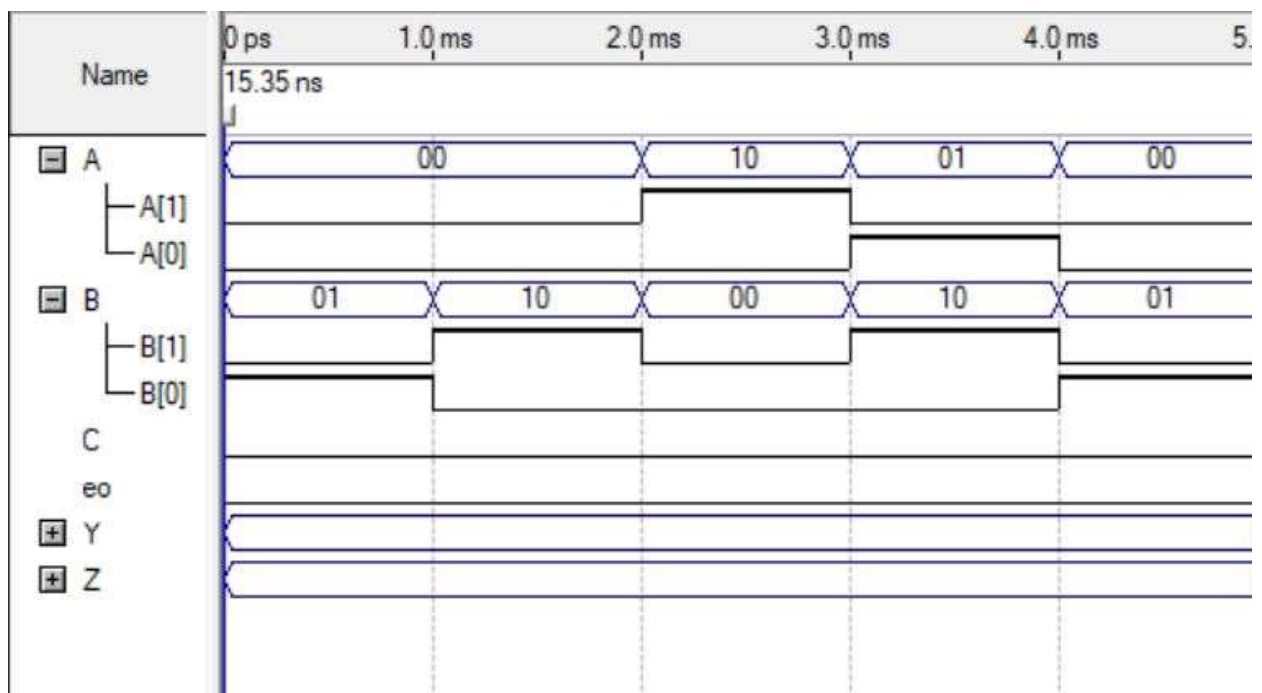
end .....
```

- 1- **(10%)** Indique entradas y salidas del circuito y su número de bits.
  
  
  
  
  
  
  
  
  
  
- 2- **(20%)** Complete en el código las zonas con puntos suspensivos.
  
  
  
  
  
  
  
  
  
  
- 3- **(15%)** Explique para qué sirven y porqué se utilizan en el código las funciones unsigned() y std\_logic\_vector().

4- (25%) Dibuje el esquemático del proceso 2. ¿Qué función realiza la entrada C en este proceso?

5- (30%) Dibuje las salidas e0, Z e Y en el siguiente cronograma.

Nota: El valor de C es cero lógico en todo el cronograma.



NOMBRE Y APELLIDOS:



Electrónica Digital. GICME GIT  
Primer Parcial. Curso 2021-2022  
27 de Octubre de 2021

<b>Problema 3 (4 puntos)</b>	40 minutos
------------------------------	------------

Se quiere diseñar un circuito lógico digital que simule el protocolo de actuación frente al COVID 19 con el propósito de entrenar a un grupo de estudiantes de primaria.

Como entradas al circuito se tomarán los síntomas experimentados por el paciente y como salidas la actuación sanitaria correspondiente en cada caso, tal y como se describe a continuación.

El simulador funcionará de la forma siguiente:

- Si el paciente no ha sido vacunado ( $V=0$ ) y presenta tos seca ( $TS=1$ ) deberá mantener la distancia social ( $DS=1$ ). Si ya ha sido vacunado ( $V=1$ ), no se recomienda acción alguna.
- Si el paciente no ha sido vacunado ( $V=0$ ) y presenta fiebre ( $FB=1$ ) deberá mantener la distancia social ( $DS=1$ ). Si ya ha sido vacunado ( $V=1$ ), no se recomienda acción alguna.
- Si el paciente presenta tos seca y fiebre conjuntamente, deberá guardar cuarentena domiciliaria ( $CD=1$ ) y distanciamiento social ( $DS=1$ ).
- Si el paciente no ha sido vacunado ( $V=0$ ) y experimenta tos seca ( $TS=1$ ) y dificultad para respirar ( $DR=1$ ) deberá acudir urgentemente al hospital ( $H=1$ ). En el caso de haber sido vacunado ( $V=1$ ) deberá acudir a su centro de salud ( $CS=1$ ) y mantener la distancia social ( $DS=1$ ).
- Si el paciente no ha sido vacunado ( $V=0$ ) y presenta fiebre ( $FB=1$ ) y dificultad para respirar ( $DR=1$ ), deberá acudir urgentemente al hospital ( $H=1$ ). En el caso de haber sido vacunado ( $V=1$ ) deberá acudir a su centro de salud ( $CS=1$ ) y mantener la distancia social ( $DS=1$ ).
- Si el paciente presenta tos seca ( $TS=1$ ), dificultad para respirar ( $DR=1$ ) y fiebre ( $FB=1$ ), haya o no sido vacunado, debe acudir urgentemente al hospital ( $H=1$ ).
- En el caso en que el paciente experimente dificultad para respirar ( $DR=1$ ), haya o no recibido la vacuna, y siempre que no experimente ningún otro síntoma, deberá acudir a su centro de salud ( $CS=1$ ) y mantener la distancia social ( $DS=1$ ).
- Para cualquier otra combinación de síntomas no mencionados, el paciente debe guardar cuarentena domiciliaria ( $CD=1$ ) y distanciamiento social ( $DS=1$ ). En caso de no presentar ninguno de los síntomas descritos no se tomará ninguna acción.

**Se pide:**

1. **(10%)** Describa en VHDL la **entidad** del circuito lógico, indicando todas las entradas y salidas y el número de bits de cada una de las señales.

NOMBRE Y APELLIDOS:

2. (30%) Rellene la tabla de verdad del sistema.

V	DR	FB	TS	DS	CD	CS	H

3. (10%) Escriba la expresión simplificada de la salida **CD** en la primera forma canónica.



NOMBRE Y APELLIDOS:

4. **(50%)** Describa en VHDL la arquitectura que implementa la funcionalidad del sistema mediante sentencias secuenciales.

**P1 Solución:**

1- Transforme 0xAB a

- a. Decimal  
0xAB=171'd
- b. Binario  
0xAB=10101011'b
- c. Octal  
0xAB=253'O
- d. BCD  
0xAB=0001 0111 0001 BCD

2- Siendo  $A=+64_{10}$  y  $B=-120_{10}$

- a. Obtenga A y B en Ca2 con 8 bits.  
A=01000000 ca2  
B= not(01111000) +1= 10001000 ca2
- b. Realice las siguientes operaciones en Ca2 con 8 bits. Indique si hay acarreo y overflow. Justifique su respuesta.
  - i. A+B  
64-120=-56 (no hay overflow)  
A= 01000000  
B= 10001000  
11001000 =-56  
No hay acarreo
  - ii. A-B  
64+120=184 (hay overflow, máximo número representable +127)  
  
A= 01000000  
-B= 01111000  
10111000, resultado con signo negativo y los dos operandos positivos- > overflow  
No hay acarreo

**Problema 2 (4 puntos)**

Para el siguiente circuito descrito en VHDL:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity prob2 is
..... PORT .....(
    A,B:   IN std_logic_vector(1 downto 0);
    C:     IN std_logic;
    Z:     OUT std_logic_vector (1 downto 0);
    Y:     OUT std_logic_vector (1 downto 0);
    eo:    OUT std_logic;
end prob2;

architecture a_func of ..... prob2 ..... is
    signal s1: std_logic_vector (..... 1 ..... downto 0);
    signal s2: unsigned (1 downto 0);
    signal s3: std_logic_vector(..... 3 ..... downto 0);

begin

    s1<=A xor B;
    s2<= unsigned(A)+ unsigned(B);
    s3<=A&B;

    p1: process(..... C,s1,s2 ..... )
        begin
            if C='0' then
                Y<=s1;
            else
                Y<=std_logic_vector(s2);
            end if;
        end process;

```

```
p2:  process(.....C,s3.....)
      begin
          e0<='0';
          if C='1' then
              Z<="00";
          else
              case s3 is
                  when "0001" =>Z<="00"; e0<='1';
                  when "0010" =>Z<="01";
                  when "0100" =>Z<="10";
                  when "1000" =>Z<="11";
                  when others => Z<="00";
              end case;
          end if;
      end process;

end .....a_func.....;
```

1- (10%) Indique entradas y salidas del circuito y su número de bits.

- Entradas:
  - A y B: 2 bits
  - C: 1 bit
- Salidas:
  - Z e Y: 2 bits
  - E0: 1 bit

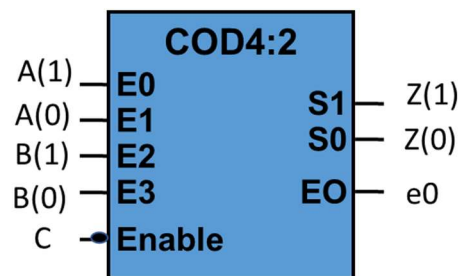
2- (20%) Complete en el código las zonas con puntos suspensivos.

3- (15%) Explique para qué sirven y porqué se utilizan en el código las funciones unsigned() y std\_logic\_vector().

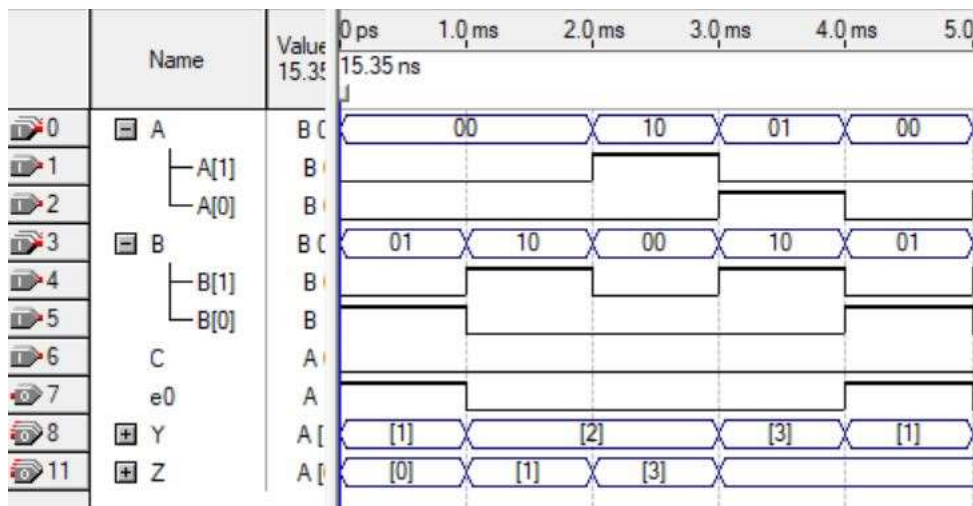
En el código se realiza una suma aritmética y los operandos son las entradas A y B de tipo std\_logic, mediante las funciones se transforman los operandos a tipo unsigned para realizar la operación y se vuelve a transformar el resultado a tipo std\_logic compatible con la salida Y.

- 4- (25%) Dibuje el esquemático del proceso 2. ¿Qué función realiza la entrada C en este proceso?

Proceso 2 implementa un codificador con enable activo a nivel bajo (Señal C). Entradas y salidas son activas a nivel alto.



- 5- (30%) Dibuje las salidas e0, Z e Y en el siguiente cronograma.



**P3 Solución**

a)

ENTITY P3 IS -- El circuito tiene 4 entradas de un bit y 4 salidas de un bit

PORT (

V, DR, FB, TS: IN STD\_LOGIC ;

DS, CD, CS, H: OUT STD\_LOGIC );

END P3;

b)

V	DR	FB	TS	DS	CD	CS	H
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	1
1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	1	0	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	0
1	1	1	0	1	0	1	0
1	1	1	1	0	0	0	1

c)

$$CD = \bar{V} \bar{DR} FB TS + V \bar{DR} FB TS = \bar{DR} FB TS (\bar{V} + V) = \bar{DR} FB TS$$

d)

ARCHITECTURE funcional OF P3 IS

Signal entrada,salida: STD\_LOGIC\_VECTOR (3 downto 0);

BEGIN

entrada&lt;= V&amp;DR&amp;FB&amp;TS;

PROCESS (entrada)

```
BEGIN
    CASE entrada is
        WHEN "0001"|"0010"=> salida<="1000";
        WHEN "0011"|"1011"=> salida<="1100";
        WHEN "0100"|"1100"|"1101"|"1110"=> salida<="1010";
        WHEN "0101"|"0110"|"0111"|"1111"=> salida<="0001";
        WHEN OTHERS => salida<="0000";
    END CASE;
END PROCESS;
H<= salida(0);
CS<= salida(1);
CD<= salida(2);
DS<= salida(3);

end funcional;
```