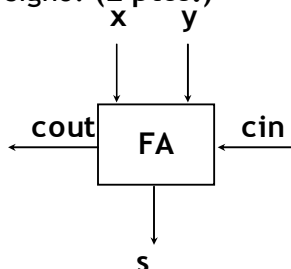


PRUEBA DE ENTRADA

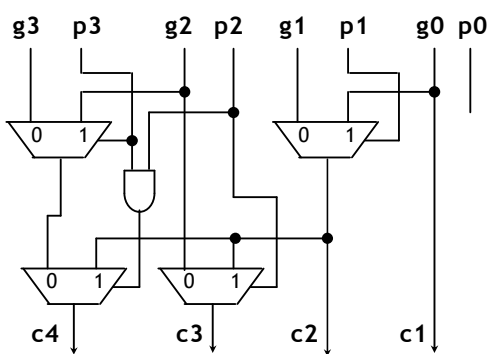
NOMBRE : _____ FECHA: ____/____/2005

CÓDIGO : _____ LAB. N°: 4 HORARIO: H-441

1. Con sumadores completos de 1 bit, elabore un circuito que obtenga el valor absoluto de un número en complemento a 2 de 4 bits. La salida tendrá 4 bits sin signo. (2 ptos.)



2. Complete el código VHDL para el circuito mostrado. Use la Descripción Estructural. Asuma que ya existe el bloque MUX 2 a 1 ('m21.vhd'). (2 ptos.)



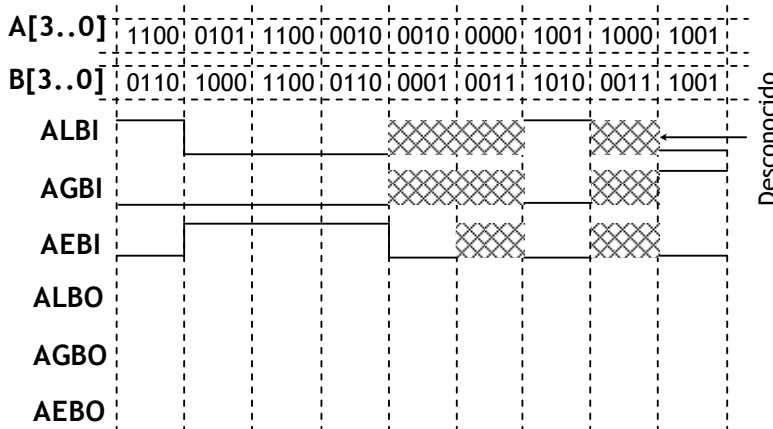
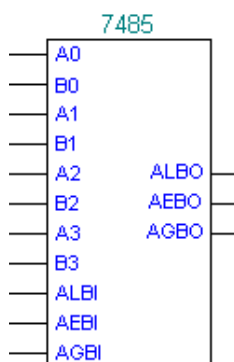
```
library ieee;
use ieee.std_logic_1164.all;
use work.dig_pack.all;
entity cp is
port(g3,g2,g1,g0: in std_logic;
p3,p2,p1,p0: in std_logic;
c1,c2,c3,c4:buffer std_logic);
end cp;
```

```
architecture est of cp is
signal xx, yy, cx: std_logic;
begin
```

```
library ieee;
use ieee.std_logic_1164.all;
package dig_pack is
component m21 is
port(a,b,s: in std_logic;
f : out std_logic);
end component;
end dig_pack;
```

```
end est;
```

3. Complete el diagrama de tiempos para el CI 7485: (1 pto.)

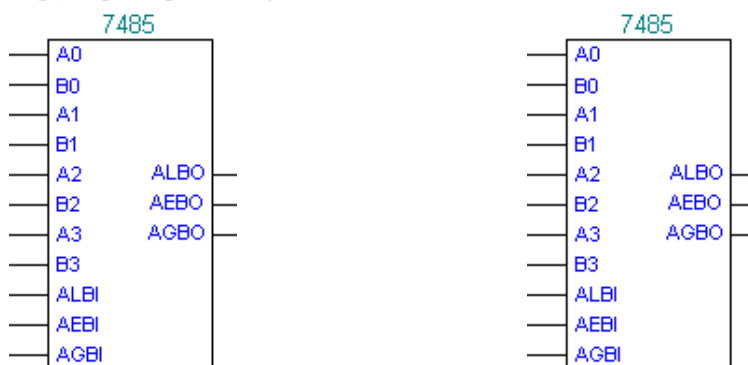


PRUEBA DE ENTRADA

NOMBRE : _____ FECHA: ____/____/2005

CÓDIGO : _____ LAB. N°: 4 HORARIO: H-442

1. En base a sólo 2 CIs 7485 implemente un comparador de 2 números positivos de 8 bits (A[7..0] y B[7..0]). Indique todas las conexiones necesarias (2 ptos.)



2. Elabore el código VHDL de un sumador de 2 números de 3 bits en complemento a 2. Para considerar todos los casos posibles, la salida debe tener 4 bits en complemento a 2 (no hay acarreo de salida ni de entrada). Utilice la descripción estructural, asuma que ya existe el sumador completo: 'fa.vhd' (2 ptos.)

```

library ieee;
use ieee.std_logic_1164.all;

package dig_pack is

    component fa is
        port(
            cin,x,y: in std_logic;
            s,cout : out std_logic);
        end component;

end dig_pack;

library ieee;
use ieee.std_logic_1164.all;
use work.dig_pack.all;

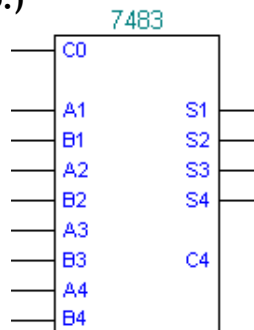
entity sum3 is
    port (
        a, b: in std_logic_vector(2 downto 0);
        s : out std_logic_vector(3 downto 0));
end sum3;

architecture est of sum3 is

begin

end est;
    
```

3. Implemente un circuito que obtenga el valor absoluto de un número de 4 bits en complemento a 2 (N[3..0]). La salida será un número positivo de 4 bits. Dispone de un CI 7483 y compuertas lógicas. (1 pto.)



PRUEBA DE ENTRADA

NOMBRE : _____ FECHA: ____/____/2005

CÓDIGO : _____ LAB. N°: 4 HORARIO: H-443

1. En base sólo a MUXs 2 a 1, implemente un MUX 4 a 1 en código VHDL. Asuma que ya existe el código para el MUX 2 a 1 (m21.vhd). (2 ptos.)

```

library ieee;
use ieee.std_logic_1164.all;

package dig_pack is
  component m21 is
    port(
      a, b, s: in std_logic;
      f : out std_logic);
  end component;
end dig_pack;

library ieee;
use ieee.std_logic_1164.all;
use work.dig_pack.all;

entity mux41 is
  port ( a, b ,c, d: in std_logic;
        s : in std_logic_vector(1 downto 0);
        f : out std_logic);
end mux41;

architecture est of mux41 is
begin

end est;

```

2. Implemente el circuito descrito mediante el siguiente código VHDL. Use el MUX 2 a 1 ('m21.vhd') como bloque. (2 ptos.)

```

library ieee;
use ieee.std_logic_1164.all;
package dig_pack is
  component m21 is
    port(a,b,s: in std_logic;
          f : out std_logic);
  end component;
end dig_pack;
library ieee;
use ieee.std_logic_1164.all;
use work.dig_pack.all;

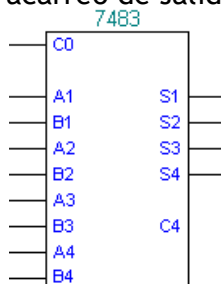
entity cp is
  port(g3,g2,g1,g0: in std_logic;
        p3,p2,p1,p0: in std_logic;
        c1,c2,c3,c4:buffer std_logic);
end cp;

architecture est of cp is
  signal yy, cx: std_logic;
begin

  yy <= p3 and p2; c1 <= g0;
  t0: m21 port map (a=>g1, b=>g0, s=>p1,
                    f=>c2);
  t1: m21 port map (a=>g3, b=>g2, s=>p3,
                    f=>cx);
  t2: m21 port map (a=>cx, b=>c2, s=>yy,
                    f=>c4);
  t3: m21 port map (a=>g2, b=>c2, s=>p2,
                    f=>c3);
end est;

```

3. Implemente, con sólo el CI 7483, un sumador de 2 números de 3 bits en complemento a 2 (a[2..0] y b[2..0]). La salida debe tener 4 bits en complemento a 2 (s[3..0]), no hay acarreo de salida ni de entrada. (1 pto.)

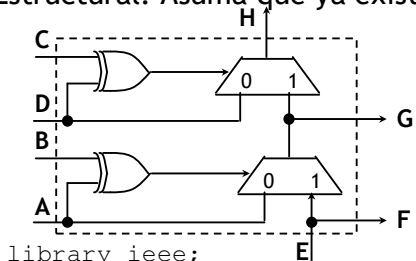


PRUEBA DE ENTRADA

NOMBRE : _____ FECHA: ____/____/2005

CÓDIGO : _____ LAB. N°: 4 HORARIO: H-444

1. Complete el código VHDL para el circuito mostrado. Use la Descripción Estructural. Asuma que ya existe el bloque MUX 2 a 1 ('m21.vhd'). (2 pts.)



```
library ieee;
use ieee.std_logic_1164.all;
package dig_pack is
  component m21 is
    port(a,b,s: in std_logic;
         f : out std_logic);
  end component;
end dig_pack;
library ieee;
use ieee.std_logic_1164.all;
use work.dig_pack.all;
```

```
entity cp is
port( a, b, c, d, e: in std_logic;
      f, g, h: buffer std_logic);
end cp;

architecture est of cp is

begin

end est;
```

2. Complete el diagrama de tiempos del circuito cuya descripción en VHDL se muestra. 'fa.vhd' es un sumador completo de 1 bit: (2 pts.)

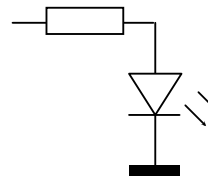
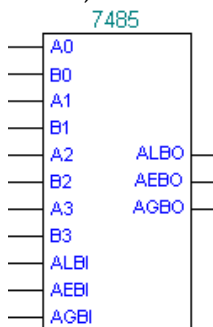
```
library ieee;
use ieee.std_logic_1164.all;
package dig_pack is
  component fa is
    port(cin,x,y : in std_logic;
         s,cout: out std_logic);
  end component;
end dig_pack;
library ieee;
use ieee.std_logic_1164.all;
use work.dig_pack.all;

entity circ is
port (
  a,b: in std_logic_vector(2 downto 0);
  s: out std_logic_vector(3 downto 0));
end circ;
architecture est of circ is
  signal c:std_logic_vector(1 to 3);
```

```
signal zero: std_logic;
begin
  zero <= '0';
  t0: fa port map (cin=>zero, x=>a(0),
                  y=>b(0), s=>s(0), cout=>c(1));
  t1: fa port map (cin=>c(1), x=>a(1),
                  y=>b(1), s=>s(1), cout=>c(2));
  t2: fa port map (cin=>c(2), x=>a(2),
                  y=>b(2), s=>s(2), cout=>c(3));
  t3: fa port map (cin=>c(3), x=>a(2),
                  y=>b(2), s=>s(3));
end est;
```

A[2..0]	000	110	011	111	100	001
B[2..0]	111	101	001	101	010	011
S[3..0]						

3. En base al CI 7485 implemente un circuito que encienda un LED cuando la temperatura (codificada con 4 bits) sea menor que el valor '0011'. (1 pto.)

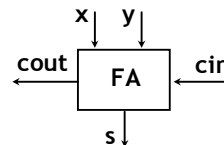


PRUEBA DE ENTRADA

NOMBRE : _____ FECHA: ____/____/2005

CÓDIGO : _____ LAB. N°: 4 HORARIO: H-445

1. Se muestra cómo se realiza la operación de multiplicación de 2 números de 2 bits sin signo. Elabore el circuito que realiza esta multiplicación en base a sumadores completos de 1 bit y compuertas lógicas: (2 ptos.)



$$\begin{array}{r}
 a_1 \ a_0 \ x \\
 b_1 \ b_0 \\
 \hline
 a_1b_0 \ a_0b_0 \\
 a_1b_1 \ a_0b_1 \\
 \hline
 m_3 \ m_2 \ m_1 \ m_0
 \end{array}$$

2. Complete el diagrama de tiempos del circuito cuya descripción en VHDL se muestra. 'fa.vhd' es un sumador completo de 1 bit: (2 ptos.)

```

library ieee;
use ieee.std_logic_1164.all;
package dig_pack is
  component fa is
    port( cin, x, y : in std_logic;
          s, cout : out std_logic);
  end component;
end dig_pack;
library ieee;
use ieee.std_logic_1164.all;
use work.dig_pack.all;

entity circ is
  port (
    a: in std_logic_vector(3 downto 0);
    r: out std_logic_vector(3 downto 0));
end circ;

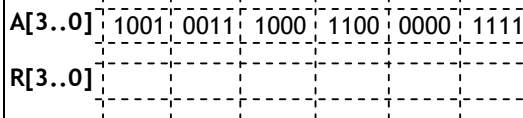
architecture est of circ is
  signal c: std_logic_vector(1 to 3);
  signal z: std_logic_vector(3 downto 0);
  signal zero: std_logic;

```

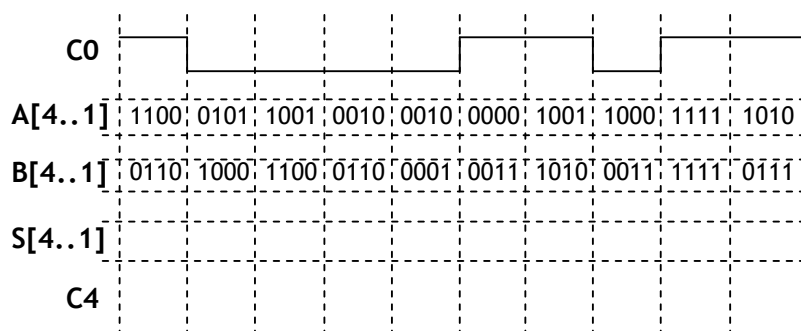
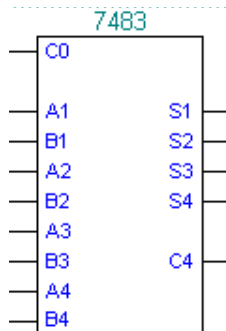
```

begin
  zero <= '0';
  z <= a xor a(3)&a(3)&a(3)&a(3);
  m: fa port map (cin=>a(3), x=>zero,
                 y=>z(0), s=>r(0), cout=>c(1));
  n: fa port map (cin=>c(1), x=>zero,
                 y=>z(1), s=>r(1), cout=>c(2));
  o: fa port map (cin=>c(2), x=>zero,
                 y=>z(2), s=>r(2), cout=>c(3));
  p: fa port map (cin=>c(3), x=>zero,
                 y=>z(3), s=>r(3));
end est;

```



3. Complete el diagrama de tiempos para el CI 7483: (1 pto.)

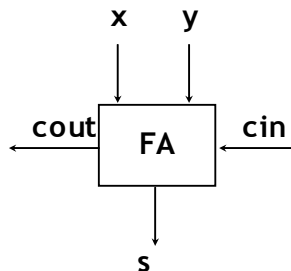


PRUEBA DE ENTRADA

NOMBRE : _____ FECHA: ____/____/2005

CÓDIGO : _____ LAB. N°: 4 HORARIO: H-446

-
1. En base al sumador completo de 1 bit, elabore el circuito para un sumador de 2 números de 3 bits en complemento a 2. Para considerar todos los casos posibles, la salida debe tener 4 bits en complemento a 2 (no hay acarreo de salida ni de entrada en el bloque final) . (2 pts.)



2. Realice el código VHDL para un circuito que obtenga el valor absoluto de un número en complemento a 2 de 4 bits en base a sumadores completos de 1 bit. La salida tendrá 4 bits sin signo. Utilice la descripción estructural, para esto asuma que ya existe el sumador completo: 'fa.vhd'. (2 pts.)

```

library ieee;
use ieee.std_logic_1164.all;

package dig_pack is

    component fa is
        port(
            cin,x,y: in std_logic;
            s,cout : out std_logic);
        end component;

end dig_pack;

library ieee;
use ieee.std_logic_1164.all;
use work.dig_pack.all;

entity vabs is
    port (
        a: in std_logic_vector(2 downto 0);
        f : out std_logic_vector(3 downto 0));
end vabs;

architecture est of vabs is

begin

end est;
    
```

3. Complete la tabla de verdad del CI 7485, en donde además de los números de entrada 'A' y 'B' de bits cada una, se dispone de las entradas de expansión (entradas en cascada) ALBI, AGBI, AEBI. (1 pto.)

A3, B3	A2,B2	A1,B1	A0,B0	AGBI	ALBI	AEBI	AGBO	ALBO	AEBO
A3>B3	X	X	X	X	X	X			
A3<B3	X	X	X	X	X	X			
A3=B3	A2>B2	X	X	X	X	X			
A3=B3	A2<B2	X	X	X	X	X			
A3=B3	A2=B2	A1>B1	X	X	X	X			
A3=B3	A2=B2	A1<B1	X	X	X	X			
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X			
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X			
A3=B3	A2=B2	A1=B1	A0=B0	1	0	0			
A3=B3	A2=B2	A1=B1	A0=B0	0	1	0			
A3=B3	A2=B2	A1=B1	A0=B0	0	0	1			