Condiciones de entrega:

Horario y duración:

a. La hora de inicio es 8.30 hs y la de finalización es 11.30hs
b. Todas las entregas realizadas fuera de término no serán tenidas en cuenta para su corrección

Entrega:

a. Se utilizará el repositorio individual utilizado para la entrega de los trabajos prácticos
b. Todos los archivos .vhd y resoluciones de los ejercicios junto con el tema del parcial deben ser subidos a una carpeta con el nombre primerParcial. El nombre de los archivos vhd deben coincidir con el nombre de la entidad.
c. Se recomienda hacer un commit cada 30 minutos.
d. Al finalizar el parcial debe hacer el commit al repositorio.

Condiciones generales

a. Cualquier diseño que infiera un latch o tenga errores de síntesis será invalido.
b. Todos los diseños se realizan con el siguiente dispositivo xc7z010clg400-1
c. Implemente un solo proyecto para todos los ejercicios.
d. Todos los ejercicios deben tener su testbench que demuestre su funcionamiento.
e. Serán mejor considerados los ejercicios que utilicen menos recursos.
f. Todos los resets son sincrónicos activos alto.

Parte práctica:

1. (2 Puntos)Implemente un circuito que reciba un número signado en CA2 de N_IN bits y sature el mismo a N_OUT bits. La entidad es la siguiente

```
entity sat is
  Generic (N_IN: integer := 16;
           N_OUT: integer := 8);
  Port ( entrada : in std_logic_vector (N_IN - 1 downto 0);
         salida : out std_logic_vector (N_OUT - 1 downto 0));
end sat;
```

Se pide:

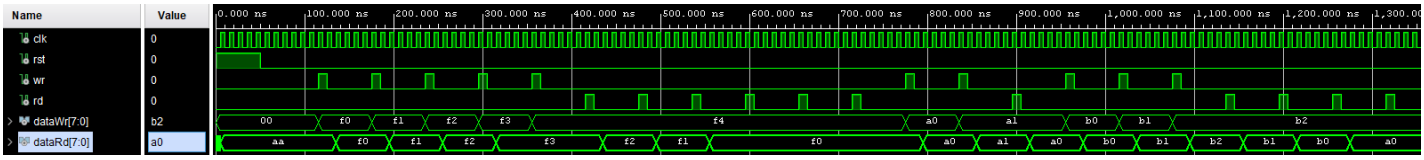
a. Un diagrama en bloques del VHDL implementado
b. La descripción en VHDL.

2. (3 Puntos)Implementar una LIFO (Último en entrar, primero en salir) utilizando una blockram de 4 posiciones de 8 bits. Este bloque permitirá ir escribiendo datos en la block ram y leyéndolos en orden inverso al ingresado, en caso de activarse la lectura (rd) y escritura (wr) a la vez se le dará prioridad a la escritura. En el port de lectura siempre se mostrará el último valor escrito, salvo en el estado inicial antes de escribir algún valor donde se mostrará el contenido de la posición cero de la blockram. Si se escriben más de 4 valores sin leer ninguno y se intenta escribir un quinto valor no se realizara esa última escritura, permaneciendo el port de salida sin cambios

```
entity lifo is
  Port ( clk : in std_logic;
         rst : in std_logic;
         wr : in std_logic;
         dataWr : in std_logic_vector (7 downto 0);
         rd : in std_logic;
         dataRd : out std_logic_vector (7 downto 0));
end lifo;
```

Donde

- clk: El clock del sistema
- rst: Reset sincrónico activo en alto
- dataWr: Es el port de entrada de los datos que se escribirán.
- wr: Cuando está en 1 esta entrada se escribe en la block ram dataWr, en caso de estar llena la lifo no escribe el valor.
- dataRd: Es el port de salida de datos, muestra el último valor escrito. Si la lifo está vacía muestra el contenido de la dirección cero de la blockram.
- rd: Cuando está en 1 esta entrada se lee un nuevo dato de la block ram y lo coloca en dataOut.



3. (3 Puntos)Implemente un circuito que funcione como cronómetro que dispondrá de dos entradas, start y stop. Ambas entradas son activas en el flanco ascendente de ellas mismas dado que pueden permanecer en estado alto más de un ciclo de clock. El comportamiento del circuito es el siguiente:
   a. Si se activa start el contador comienza a contar hasta que se active stop, mientras tanto se ignoran otras activaciones de la entrada start.
   b. Si el contador está contando se detendrá solo al activarse la entrada stop.
   c. Si el contador está detenido y se activa la entrada stop el contador vuelve a cero.
   d. Si el contador está detenido y se activa la entrada start el contador continúa contando.
   e. Si en alguna condición se activa start y stop a la vez en cualquier condición no se realizará ninguna acción.

   Se pide:
   a. Dibujar el diagrama de bloques del circuito y el diagrama de estados.
   b. Realizar la descripción del circuito en VHDL cuya entidad es:

```
entity cronos is
Generic (SYS_FREC: integer := 200000000;
          N: integer := 6);
Port ( clk: in std_logic;
       rst: in std_logic;
       start: in std_logic;
       stop: in std_logic;
       minutos: out std_logic_vector (N-1 downto 0);
       segundos: out std_logic_vector (N-1 downto 0));
end cronos;
```

   Donde:
   - clk: Es el clock del sistema.
   - rst: Es el reset sincrónico activo alto.
   - start y stop: Entradas de control del cronómetro.
   - minutos: Salida de minutos del cronómetro.
   - segundos: Salida de segundos del cronómetro.

   c. Determine los slack de setup y hold para una frecuencia de 200MHz. En caso de ser negativo alguno de los slacks indique el camino crítico donde ocurre y una posible solución.

4. (2 puntos)Dado el siguiente código en VHDL dibuje el circuito resultante

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity sumador is
    generic ( N : integer := 4);
    Port ( opA       : in  STD_LOGIC_VECTOR (N-1 downto 0);
           opB       : in  STD_LOGIC_VECTOR (N-1 downto 0);
           res       : out  STD_LOGIC_VECTOR (N-1 downto 0));
end sumador;

architecture Behavioral of sumador is
    signal res_S : STD_LOGIC_VECTOR (N-1 downto 0);
    signal ov_S : STD_LOGIC;
    signal s  : STD_LOGIC_VECTOR (1 downto 0);
    signal satMax, satMin : STD_LOGIC_VECTOR (N-1 downto 0);
begin

ov_S <= '1' when ((signed (opA) >= 0) and (signed (opB) > 0) and (signed(res_S) <= 0))
else    '1' when ((signed (opA) < 0)  and (signed (opB) < 0) and (signed(res_S) >= 0))
else    '0';

    res_S <= std_logic_vector (signed (opA) + signed (opB));
    s <= ov_S & res_S(N-1);
    satMax(N-1) <= '0'; satMax(N-2 downto 0) <= (others => '1');
    satMin <= not (satMax);
    with s select
        res <= res_S    when "00",
               res_S    when "01",
               satMin   when "10",
               satMax   when others;
```