

Condiciones de entrega:

Horario y duración:

- La hora de inicio es 9.00hs y la de finalización es 12.00hs
- Todas las entregas realizadas fuera de término no serán tenidas en cuenta para su corrección

Entrega:

- Se utilizará el repositorio individual utilizado para la entrega de los trabajos prácticos
- Todos los archivos .vhd junto con el tema del parcial deben ser subidos a una carpeta con el nombre 1P. El nombre del archivo debe coincidir con el nombre de la entidad.
- Cada ejercicio debe tener su testbench que demuestre el funcionamiento. El nombre de la entidad de testbench, es igual al nombre de la entidad a probar con el agregado de un "_tb"
- Se recomienda hacer un commit cada 30 minutos.
- Al finalizar el parcial debe hacer el commit al repositorio y subirlo a la tarea asignada en el campus virtual.

Parte práctica:

- (2.5 Puntos) Implemente un circuito que coloque en su salida un uno durante un ciclo de clock, cuando detecte en la entrada la secuencia "1010" que entra de forma serial por el port d. El port d deberá ser leído en el flanco ascendente del port sync (Recuerde que solo puede usar rising_edge o falling_edge sobre el port clk)

```
entity myDetector is
Port ( clk: in std_logic;
      rst: in std_logic;
      d: in std_logic;
      sync : in std_logic;
      q : out std_logic);
end myDetector ;
```

- (2.5 Puntos) Implemente un circuito que genere la siguiente secuencia 000 - 100 - 010 - 111 cada valor de la secuencia debe permanecer por 100ns. Suponga que el clock del sistema tiene un periodo de 10ns

```
entity myGenCnt is
Port ( clk: in std_logic;
      rst: in std_logic;
      q : out std_logic_vector (2 downto 0));
end myGenCnt;
```

- (2.5 Puntos) Implemente en VHDL un circuito que realice la la siguiente operación $(3 / 2) * d$. Siendo el d una magnitud. La parte entera del resultado debe mostrar el resultado correcto, mientras que la parte fraccionaria del resultado puede ignorarse No está permitido el uso de multiplicadores.

```
entity myCalc is
Generic (N : integer := 4);
Port ( d: in std_logic_vector (N - 1 downto 0);
      r: out std_logic_vector (N downto 0));
end myCalc;
```

4. (2.5 Puntos) Implemente en VHDL un circuito que reciba dos números a y b, luego realice las siguientes operaciones

- a. 00: $a + b$
- b. 01: $a - 1$
- c. 10: $|a|$
- d. 11: a and b bit a bit

```
entity myAlu is
  Generic (N : integer := 4);
  Port ( a: in std_logic_vector (N - 1 downto 0);
        b: in std_logic_vector (N - 1 downto 0);
        op: in std_logic_vector (1 downto 0);
        r: out std_logic_vector (N - 1 downto 0);
        ov : out std_logic);
end myAlu;
```

Donde:

- a y b son los operandos, son números signados.
- op: Es la operación a realizar
- r: Es el resultado de la operación
- ov: Es el overflow de la operación