



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
INSTITUTO REGIONAL BUENOS AIRES

**Ingeniería en Electrónica**  
**Técnicas Digitales I**

**Examen Final**

Fecha:

Apellido y nombre:

Leg:

Aclaraciones: No se tomarán en cuenta soluciones sin su debida justificación

La duración del examen es de 2h  
Condiciones de aprobación: 6  
Aprobado: 6 a 10 puntos  
Coloquio: 5 puntos  
Desaprobado: 0 a 4 puntos

P	A	b	c	d	*
1	2.5				
2	2				
3	1	1	1		
4	2.5				
Nota:					

1. Implemente un conversor gray a binario utilizando xor y for generate

```
entity grayBinario is
  generic (N: integer := 4);
  port ( gray : in std_logic_vector (N-1 downto 0);
        binario : out std_logic_vector (N-1 downto 0));
end grayBinario;
```

Utilice las siguientes expresiones para realizar la conversión:

- $binario(i) = gray(i);$  para  $n = N - 1$
- $binario(i) = binario(i + 1) \text{ xor } gray(i);$  para  $(N - 2) \leq i \leq 0$

2. Implemente en VHDL un circuito con la siguiente entidad que realice operaciones aritméticas y lógicas

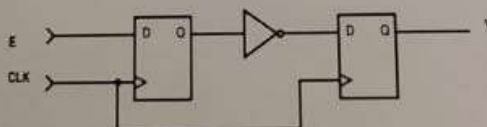
```
entity myALU is
  generic (N : integer := 8);
  port (a: in std_logic_vector (N-1 downto 0);
        b: in std_logic_vector (N-1 downto 0);
        r : out std_logic_vector (N-1 downto 0);
        ov: out std_logic;
        op: in std_logic_vector (1 downto 0));
end myALU;
```

Donde:

- a y b: son los operandos
- r: es el resultado de las operaciones.
- ov: salida que indica si hubo overflow en la operación.
- op: es la operación a realizar
  - 00: a + b signado (setea la señal de overflow)
  - 01: a - b signado (setea la señal de overflow)
  - 10: a and b bit a bit (setea overflow en cero)
  - 11: a or b bit a bit (setea overflow en cero)

No está permitido utilizar process en este ejercicio.

3. Dado el siguiente circuito



- Indique como calcularía la frecuencia máxima de clock y justifique su respuesta con un diagrama temporal.
- Indique como calcularía el slack de setup y justifique su respuesta con un diagrama temporal.
- Qué consecuencias tiene para el funcionamiento del circuito que el slack de setup sea negativo?

4. Realice una descripción en VHDL de un registro de desplazamiento con entrada serie, salida paralelo y reset sincrónico. La entidad es la siguiente.

```
entity registro is
  generic (N : integer := 8);
  port ( clk : in std_logic;
        rst : in std_logic;
        entradaSerie : in std_logic;
        salidaParalelo : out std_logic_vector (N-1 downto 0));
end registro;
```

1)

```
1 -----
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5 entity grayBinario is
6     Generic(N:integer:=4);
7     Port ( gray : in  STD_LOGIC_VECTOR (N-1 downto 0);
8           binario : out STD_LOGIC_VECTOR (N-1 downto 0));
9 end grayBinario;
10
11 architecture arch_of_grayBinario of grayBinario is
12     signal aux: std_logic_vector(N-1 downto 0);
13 begin
14     aux(N-1)<=gray(N-1);
15
16     gen: for i in N-2 downto 0 generate
17
18         aux(i)<=aux(i+1) xor gray(i);
19
20     end generate;
21
22     binario<=aux;
23 end arch_of_grayBinario;
24 -----
```

2)

```
1 -----
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4 use IEEE.numeric_std.all;
5
6 entity myAlu is
7     Generic(N:integer:=8);
8     Port ( a : in  STD_LOGIC_VECTOR (N-1 downto 0);
9           b : in  STD_LOGIC_VECTOR (N-1 downto 0);
10          r : out STD_LOGIC_VECTOR (N-1 downto 0);
11          ov : out STD_LOGIC;
12          op : in  STD_LOGIC_VECTOR (1 downto 0));
13 end myAlu;
14
15 architecture arch_of_myAlu of myAlu is
16
17     signal a_sign,b_sign: signed(N-1 downto 0);
18     signal aux_r: STD_LOGIC_VECTOR (N-1 downto 0);
19     signal ov_r,ov_s: std_logic;
20     signal aux_ov: STD_LOGIC_VECTOR(3-1 downto 0);
21     begin
22         a_sign<=signed(a);
23         b_sign<=signed(b);
24
25         aux_ov<= aux_r(N-1) & a(N-1) & b(N-1);
26
27         aux_r<= std_logic_vector(a_sign+b_sign) when op="00" else
28                std_logic_vector(a_sign-b_sign) when op="01" else
29                ( (a) and (b) ) when op="10" else
30                ( (a) or (b) )when op="11";
31
32         with aux_ov select
33         ov_r<='1' when "101",
34             '1' when "010",
35             '0' when others;
36
37         with aux_ov select
38         ov_s<='1' when "100",
39             '1' when "011",
40             '0' when others;
41
42         r<=aux_r;
43
44         ov<= ov_s when op="00" else
45             ov_r when op="01" else
46             '0' when op="10" else
47             '0' when op="11";
48
49     end arch_of_myAlu;
50 -----
```

4)

```
1 -----
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4 entity registro is
5     Generic(N:integer:=8);
6     Port ( clk : in  STD_LOGIC;
7           rst : in  STD_LOGIC;
8           entradaSerie : in  STD_LOGIC;
9           salidaParalelo : out  STD_LOGIC_VECTOR (N-1 downto 0));
10 end registro;
11
12 architecture arch_of_reg of registro is
13     signal qnext,qnow: std_logic_vector(N-1 downto 0);
14 begin
15     proceso_secuencial:process(clk)
16     begin
17         if rising_edge(clk)then
18             if rst='1' then
19                 qnow<=(others=>'0');
20             else
21                 qnow<=qnext;
22             end if;
23         end if;
24     end process;
25
26     qnext<=entradaSerie & qnow(N-1 downto 1);
27
28     salidaParalelo<=qnow;
29
30 end arch_of_reg;
31 -----
32
33
```

3) a) Se debe plantear los tiempos involucrados entre la propagación de datos de 2 flip flops (A->B) ( $T_{cqA}, T_{gateAB}, T_{slackSetup}, T_{setupB}$ ).

Donde: - $T_{cqA}$ : Propagación de la entrada a la salida en el momento que llega el flanco de clock al FFD

- $T_{gateAB}$ : Propagación entre toda la lógica secuencial entre la entrada y salida de los distintos FFD, siempre se evalúa el peor caso (camino crítico).

- $T_{slackSetup}$ : Tiempo adicional sobrante, tiempo muerto

- $T_{setupB}$ : Tiempo que el dato se debe mantener estable para que el flip flop pueda registrarlo.

La ecuación quedaba:  $T = T_{cqA} + T_{gA} + T_{slackSetup} + T_{setupB}$

Buscando la peor condición para lograr el sincronismo de datos:  $T = T_{cqAmax} + T_{gAmax} + T_{setupBmax} + T_{slackSetup}$

Se hacía tender a  $T_{slackSetup}$  a cero para no tener tiempo muerto, y la frecuencia máxima quedaba como

$$F_{max} = 1 / (T_{cqAmax} + T_{gAmax} + T_{setupBmax})$$

b)  $T_{slackSetup} \Rightarrow$  partiendo de la ecuación:  $T = T_{cqA} + T_{gA} + T_{slackSetup} + T_{setupB}$

despejamos  $T_{slackSetup}$ :  $T - (T_{cqA} + T_{gA} + T_{setupB})$

analizamos peor caso:  $T_{slackSetup} = T - (T_{cqAmax} + T_{gAmax} + T_{setupBmax})$

c) se viola el sincronismo del circuito y no se asegura el funcionamiento del mismo. Por ejemplo se podía dibujar un diagrama temporal en donde se solapaban el tiempo de gate con el tiempo de set up, por lo que el dato no era estable durante el tiempo de set up y no se aseguraba su registro.