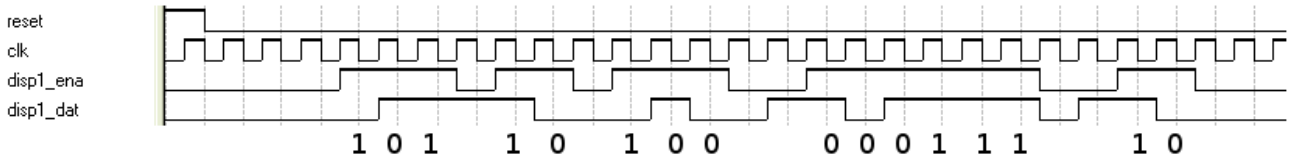
Examen Julio 2012

Se desea diseñar un circuito RTL para conectar dos dispositivos, uno que transmite datos en formato serie (***disp1***) y otro que recibe en formato paralelo (***disp2***).

disp1 transmite palabras de 8 bits en formato serie de manera síncrona utilizando dos señales (***disp1_ena*** y ***disp1_dat***). Cada vez que se produce un flanco de bajada en la señal de reloj, si la señal ***disp1_ena*** tiene valor lógico '1' significa que en la señal ***disp1_dat*** hay un bit para ser leído.

Los datos transmitidos por el dispositivo 1 son codificados de la siguiente manera: si el bit recibido es igual al anterior debe leerse como un 1 y si es diferente debe leerse como un 0. Luego de un reset, para decodificar el primer bit recibido se supone el último bit recibido fue 0. El bit más significativo es el primero en ser transmitido, ver figura.

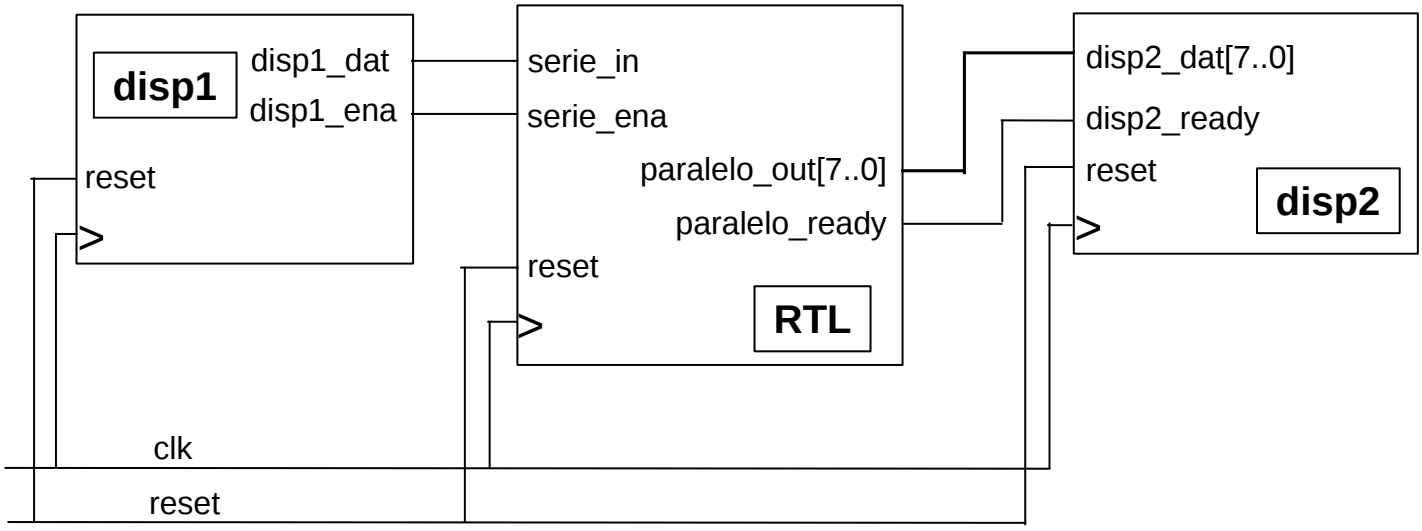
Observación: luego de un reset habrá suficiente tiempo para inicializar el circuito antes de que comience la transmisión serie.



disp2 recibe los datos en formato paralelo a través de ***disp2_dat[7..0]***, cuando la señal ***disp2_ready*** toma valor '1' el dispositivo lee la palabra de ***disp2_dat[7..0]***.

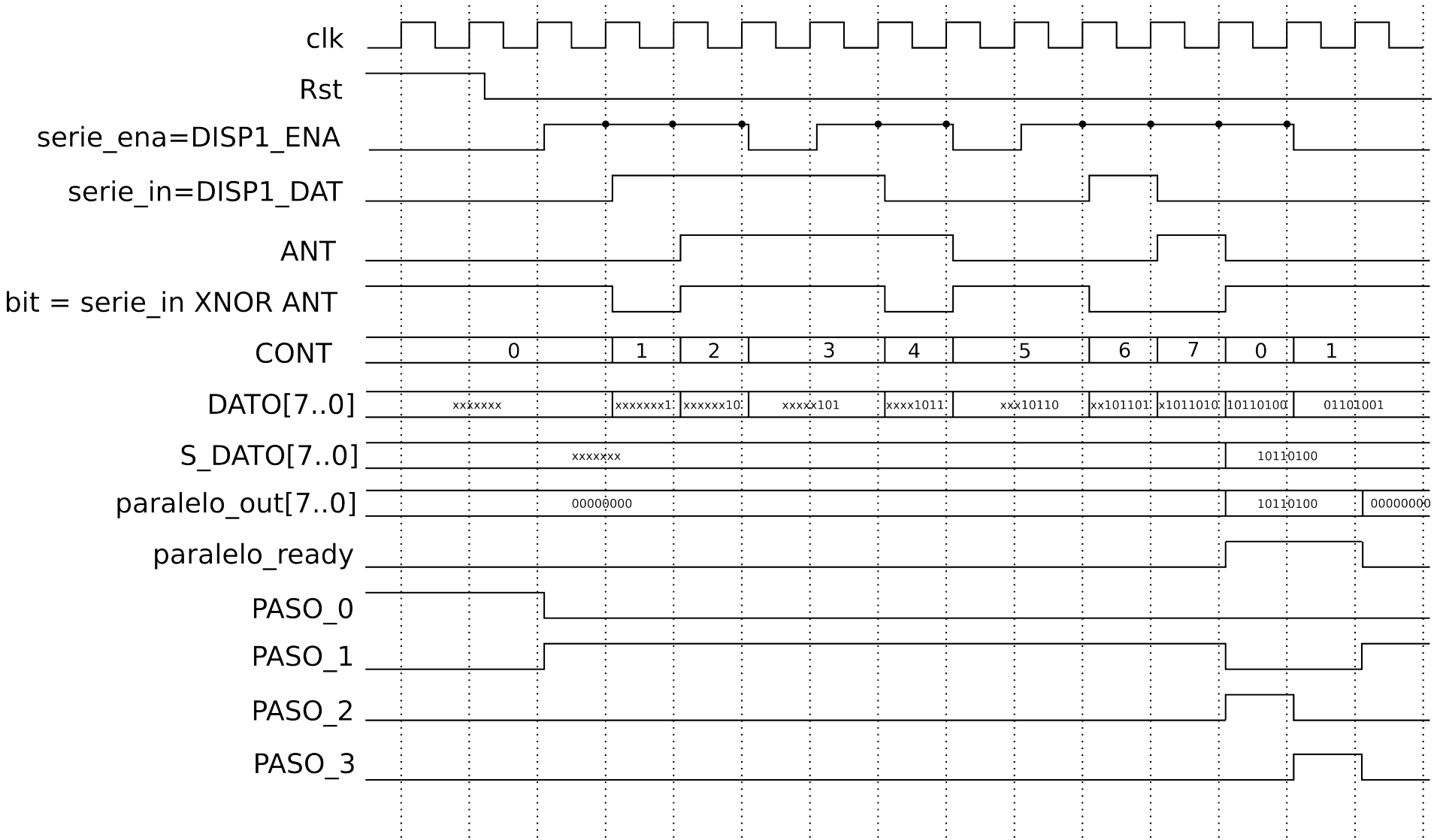
Una vez finalizada la recepción de la palabra, se deben poner durante 2 períodos de reloj la palabra decodificada en la salida **paralelo_out[7..0]** y ***disp2_ready*** en 1**.**

Observación: es posible que en la transmisión serie, el último bit de una palabra y el primero de la siguiente se transmitan en flancos de bajada consecutivos de la señal de reloj.



Se pide: **a)** Descripción RTL
b) Bloque Control
c) Bloque Datos

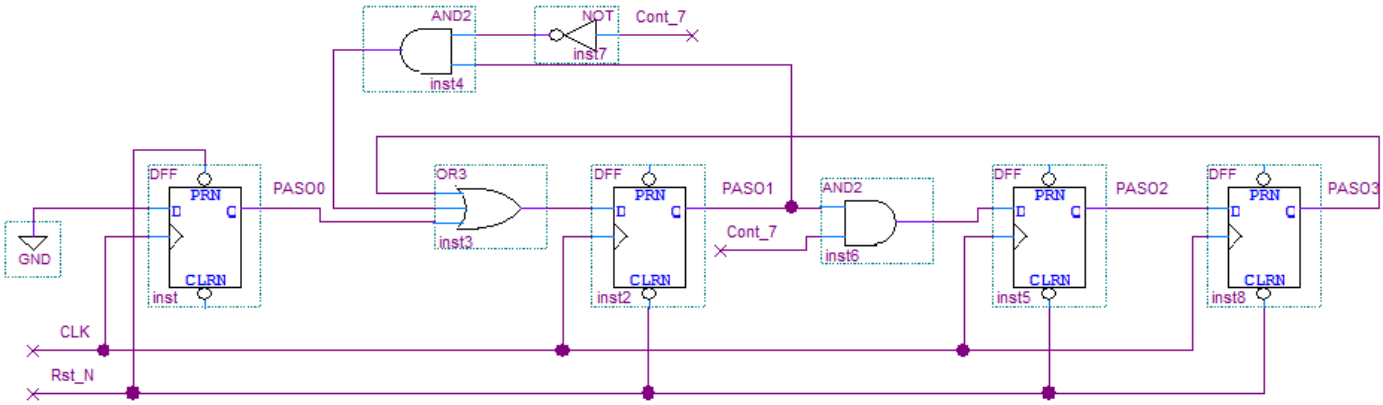
COMENTARIO:
Solución realizada en clase, se leen los bits en el flanco de subida de la señal de reloj


clk
Rst
serie_ena=DISP1_ENA
serie_in=DISP1_DAT
ANT
bit = serie_in XNOR ANT
CONT
0
1
2
3
4
5
6
7
0
1
DATO[7..0]
xxxxxxx
xxxxxxx1
xxxxxx10
xxxxx101
xxxx1011
xxx10110
xx101101
x1011010
10110100
01101001
S_DATO[7..0]
xxxxxxx
10110100
paralelo_out[7..0]
00000000
10110100
00000000
paralelo_ready
PASO_0
PASO_1
PASO_2
PASO_3

a) Descripción RTL

```
Module: Examen_Julio_2012
Input:  serie_in, serie_ena
Output: paralelo_out[8], paralelo_ready
Memory: Cont[3], Dato[8], S_Dato[8], Ant

0.
Cont[2..0] <== 000
Ant <== 0

1.
Cont[2..0] * serie_ena <== INC (Cont[2..0])
Ant * serie_ena <== serie_in
Dato[7..0] * serie_ena <== Dato[6..0], bit_in
S_Dato[7..0] * (serie_ena . Cont_7) <== Dato[6..0], bit_in
==> (Cont_7 , /Cont_7) / (2 , 1)

2.
Cont[2..0] * serie_ena <== INC (Cont[2..0])
Ant * serie_ena <== serie_in
Dato[7..0] * serie_ena <== Dato[6..0], bit_in
Paralelo_out[7..0] = S_Dato[7..0]
Paralelo_ready = 1

3.
Cont[2..0] * serie_ena <== INC (Cont[2..0])
Ant * serie_ena <== serie_in
Dato[7..0] * serie_ena <== Dato[6..0], bit_in
Paralelo_out[7..0] = S_Dato[7..0]
Paralelo_ready = 1
==>(1)

ENDSEQUENCE
CONTROLRESET(0)

Cont_7 = Cont[0].Cont[1].Cont[2]

bit_in = Ant XNOR serie_in

END
```

b) Bloque de Control



c) Bloque de Datos