

Proposeu una simulació que mostri el correcte funcionament del descodificador de dos entrades i quatre sortides descrit amb el següent codi vhdL (donat).

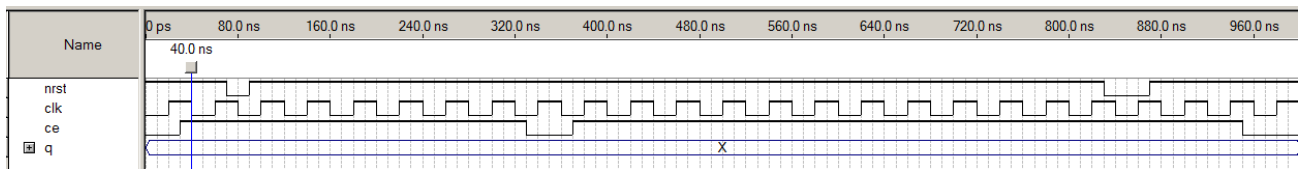
```
library ieee;
use ieee.std_logic_1164.all;

entity decodificador_4 is
port( x : in std_logic_vector(1 downto 0);
      e : in std_logic;
      y : out std_logic_vector (3 downto 0));
end decodificador_4;

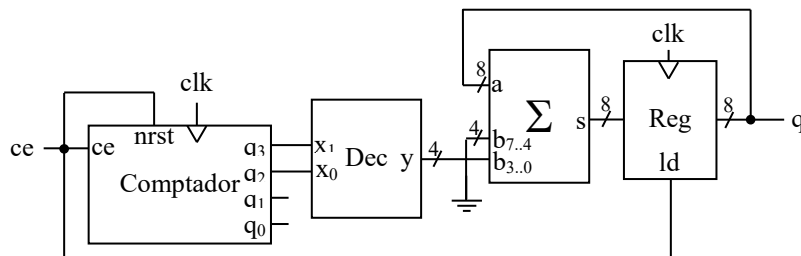
architecture arq of decodificador_4 is
begin
  y(0) <= '1' when (e='1' and x(1)='0' and x(0)='0') else '0';
  y(1) <= '1' when (e='1' and x(1)='0' and x(0)='1') else '0';
  y(2) <= '1' when (e='1' and x(1)='1' and x(0)='0') else '0';
  y(3) <= '1' when (e='1' and x(1)='1' and x(0)='1') else '0';
end arq;
```

Dissenyeu en vhdL un comptador amb topall de números de 4 bits, amb tres entrades, nrst (senyal de reset asíncron, actiu baix), clk (rellotge, sincronitza per flanc de pujada) i ce (permís per comptar), i una sortida, q, el vector d'estat del comptador. El comptador amb topall, quan arriba al valor màxim (15) el manté i no compta més fins que s'activa el reset.

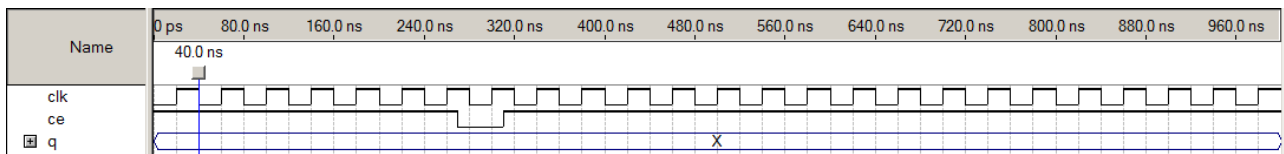
Feu-ne una simulació en que les entrades segueixin l'evolució proposada a la figura:



Finalment combineu en un sol esquemàtic els dos blocs anteriors juntament amb un sumador de 8 bits i un registre (donats) connectats com mostra la figura, per fer un comptador de pas variable:



Feu la simulació de conjunt excitant l'entrada amb el patró descrit:



Lliureu captures de pantalla de les simulacions i descripcions demanades dins d'un únic document penjant-les abans d'acabar el termini a la tasca oberta a Atenea a tal efecte.