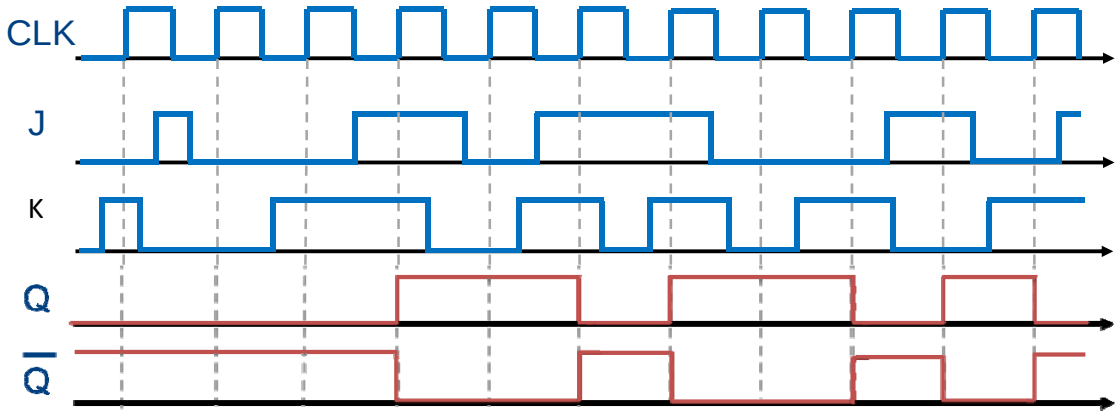
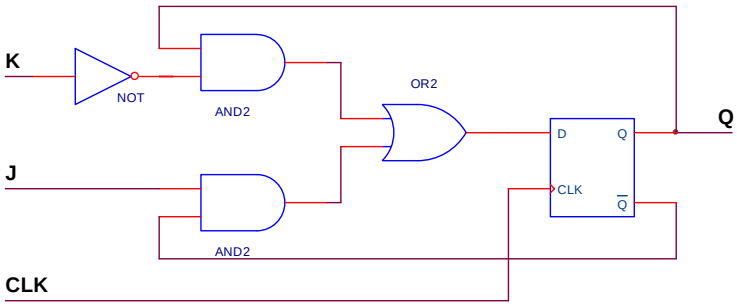
Soluciones Tema 4

Ejercicio 1.

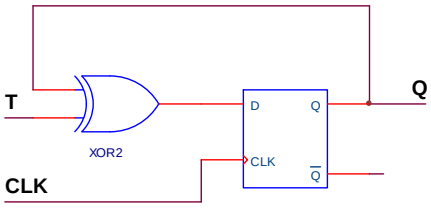


Ejercicio 2.

Biestable J-K:

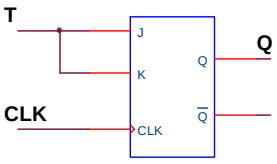


Biestable T:

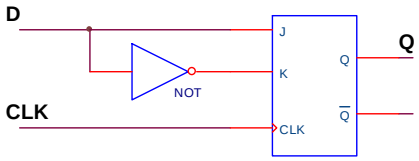


Ejercicio 3.

Biestable T:



Biestable D:

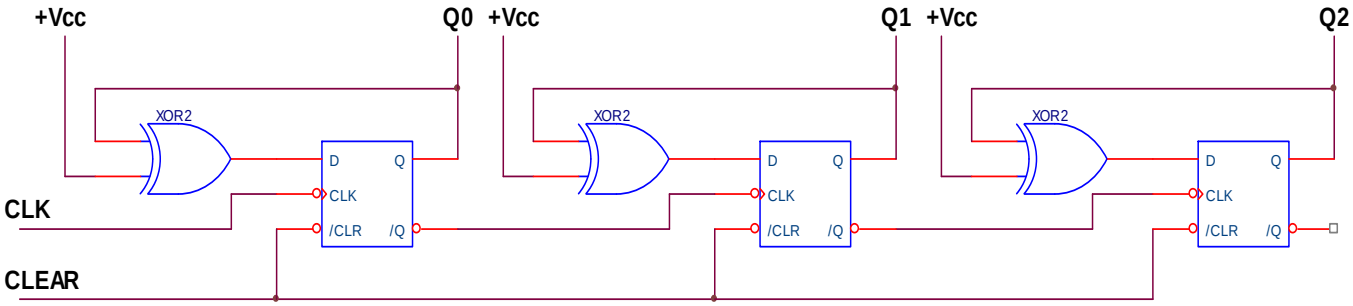


Ejercicio 4.

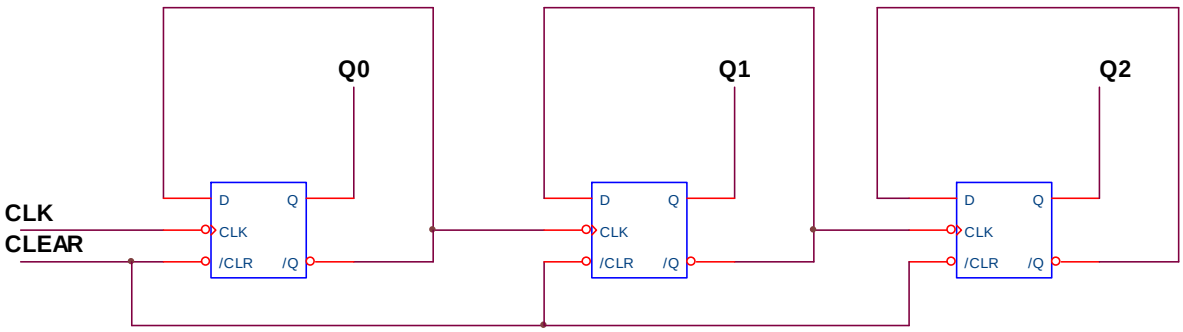
```
----------------------------------------------------------------------------
-- flipflop_D con entrada de Enable
----------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity flipflop_D is
      port(CLK,not_CLR,CE,D : in std_logic;
           Q : out std_logic);
end flipflop_D;

architecture Behavioral of flipflop_D is

begin
process (CLK,not_CLR)
begin
      if (not_CLR ='0') then Q<='0';
      elsif (CLK'event and clk='1') then
            if (CE='1')then
                  Q<=D;
            end if;
      end if;
end process;

end Behavioral;
```

Ejercicio 5.

-Contador binario asíncrono descendente de 3 bits (módulo 8).

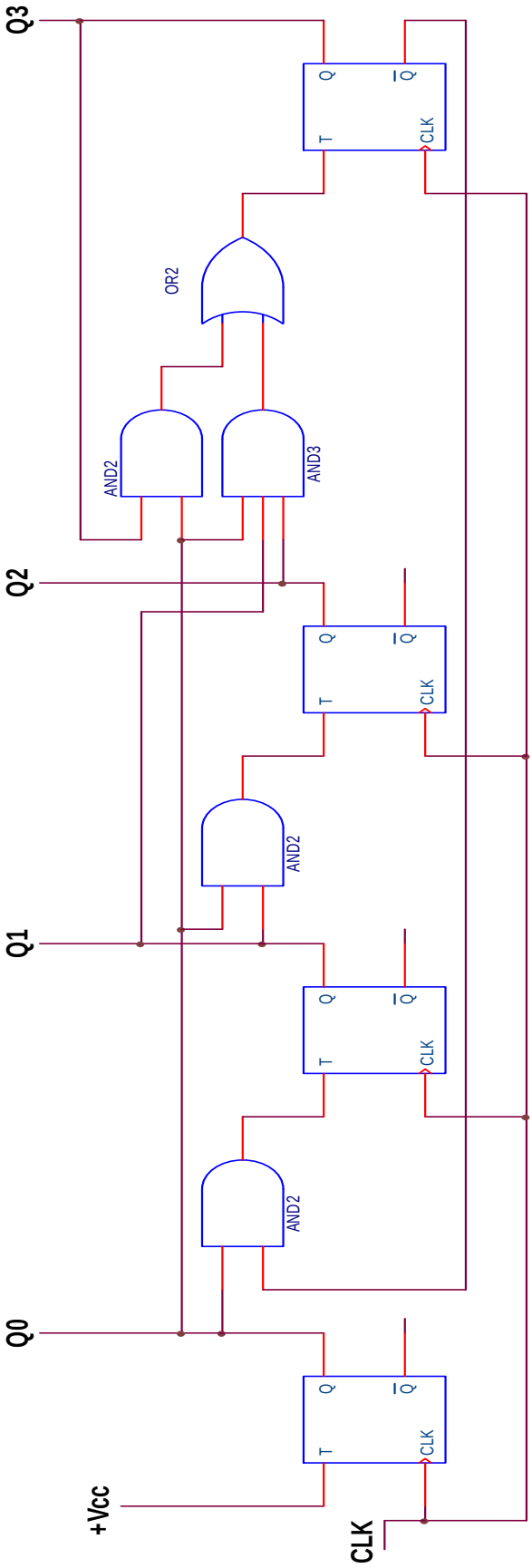


Solución óptima:



Ejercicio 6.

a)



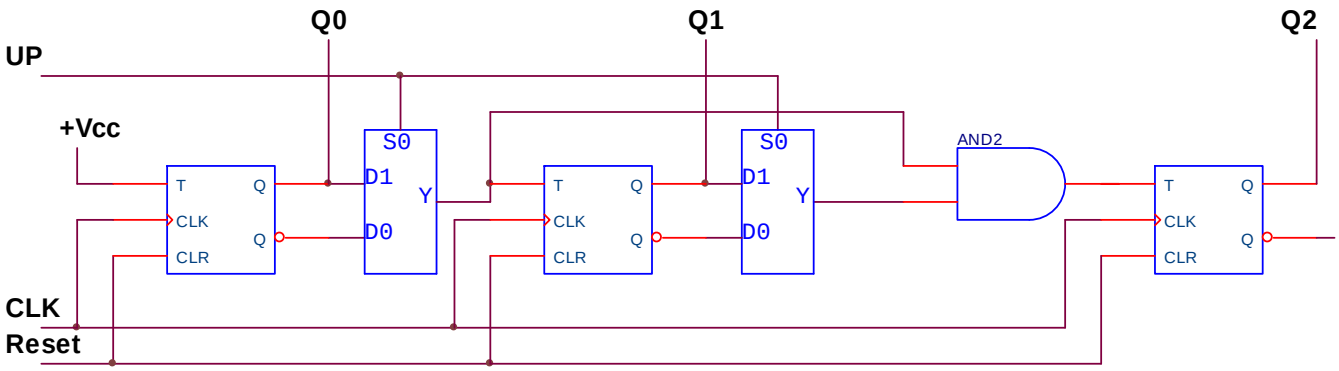
b)

```
--------------------------------------------------------------------------
-- Contador síncrono ascendente módulo 10
--------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Contador_M10 is
      Port (CLK: in std_logic;
          Reset: in std_logic;
              Q: out std_logic_vector (3 downto 0));
end Contador_M10;

architecture Behavioral of Contador_M10 is
      signal cnt_temp :STD_LOGIC_VECTOR (3 downto 0);
begin
      process(CLK,Reset)
      begin
            if(Reset='1')then
                  cnt_temp<="0000";
            elsif(CLK'event and CLK='1') then
                  if(cnt_temp="1001") then
                        cnt_temp<="0000";
                  else
                        cnt_temp<=cnt_temp+1;
                  end if;
            end if;
      end process;
      Q<=cnt_temp;
      end Behavioral;
```

Ejercicio 7.

a)

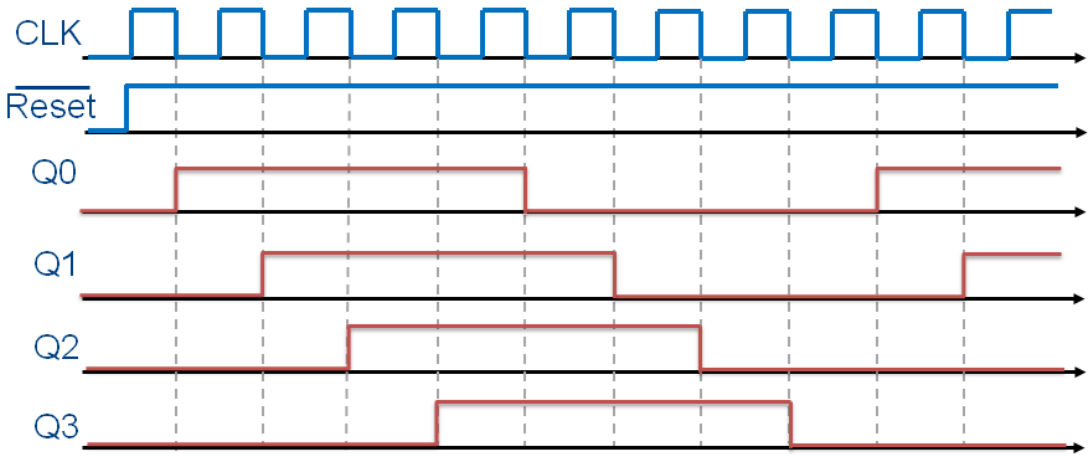
-Contador binario síncrono reversible de 3 bits (módulo 8).



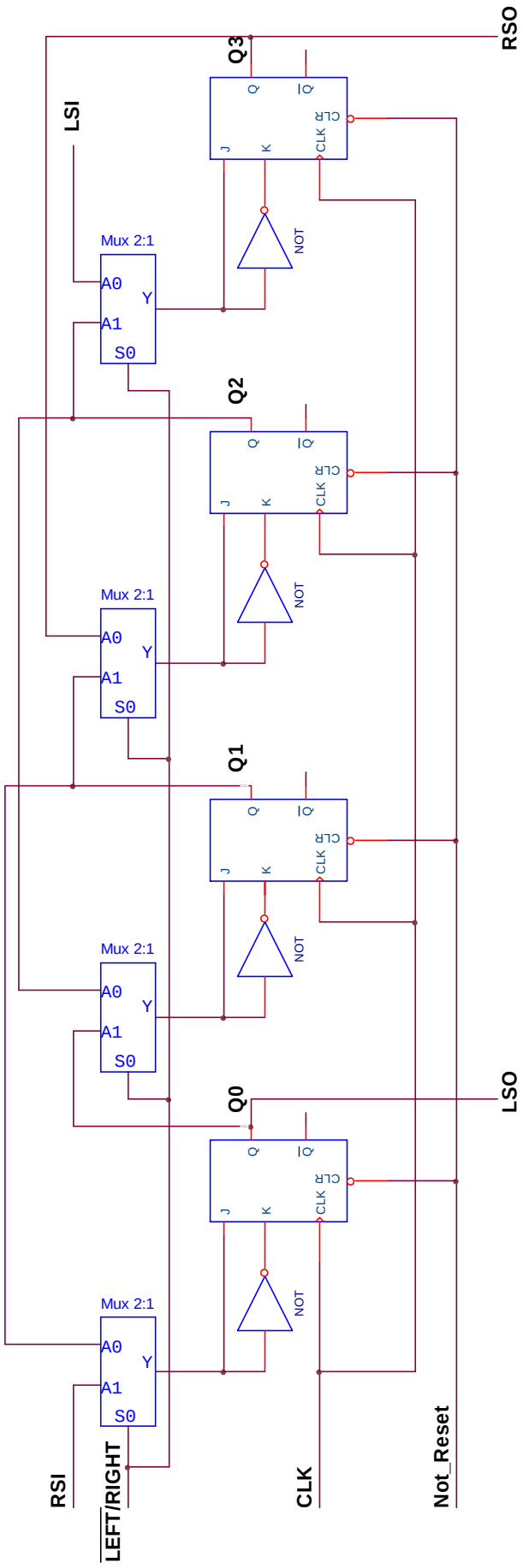
b)

```
----------------------------------------------------------------------------
-- Contador síncrono reversible módulo 8
----------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Contador_reversible is
      Port (CLK: in std_logic;
          Reset: in std_logic;
             UP: in std_logic;
              Q: out std_logic_vector (2 downto 0));
end Contador_reversible;

architecture Behavioral of Contador_reversible is
      signal cnt_temp :STD_LOGIC_VECTOR (2 downto 0);
begin
      process(CLK,Reset)
      begin
            if(Reset='1')then
                  cnt_temp<="000";
            elsif(CLK'event and CLK='1') then
                  if(UP='1') then
                        cnt_temp<=cnt_temp+1;
                  else
                        cnt_temp<=cnt_temp-1;
                  end if;
            end if;
      end process;
      Q<=cnt_temp;
end Behavioral;
```

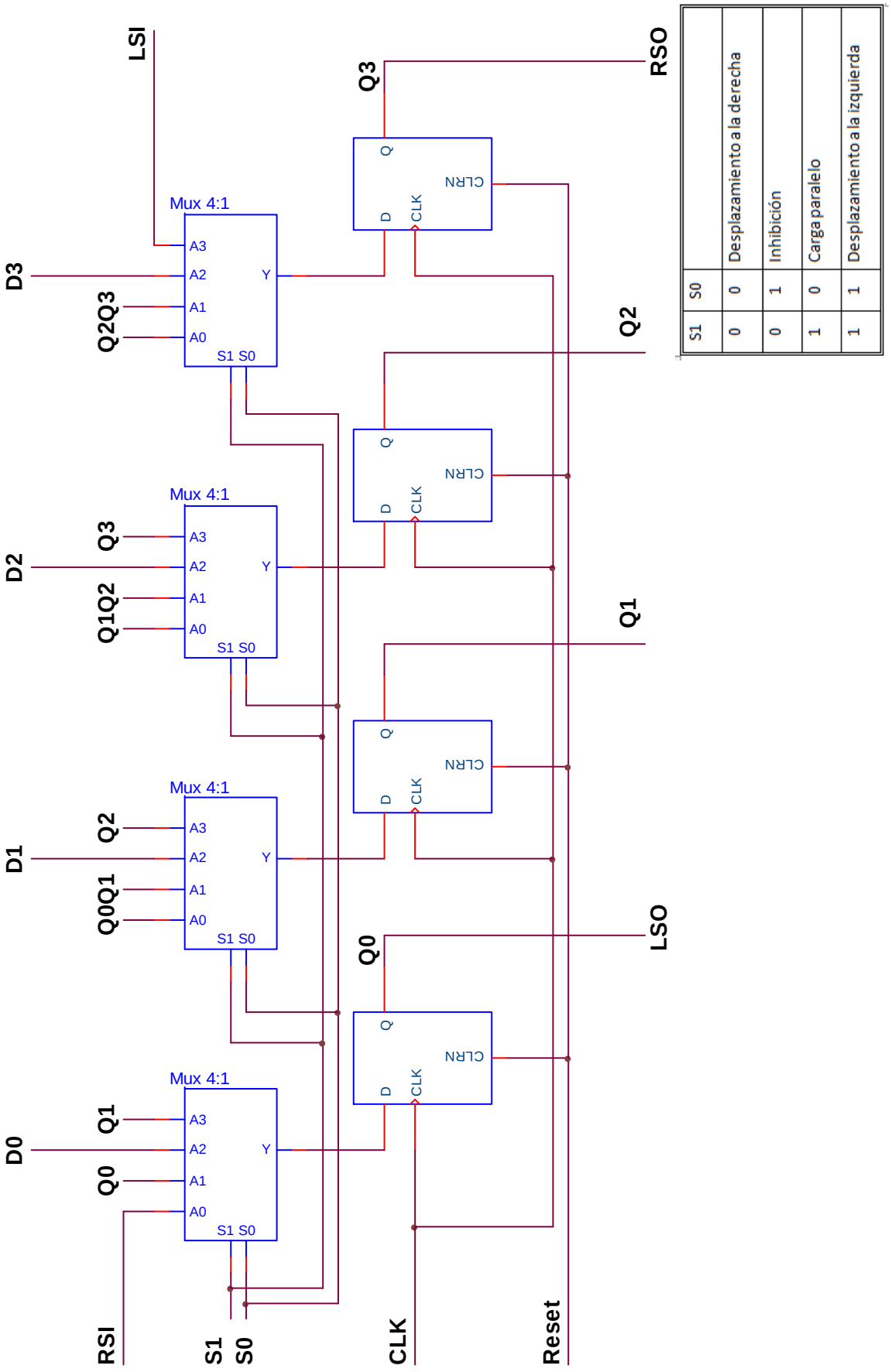
Ejercicio 8.



Ejercicio 9.


RSO
Q3
LSI
Q
Q
CLR
J
K
CLK
NOT
Mux 2:1
A0
A1
S0
Y
Q2
Q1
Q0
LSO
RSI
LEFT/RIGHT
CLK
Not_Reset

Ejercicio 10.



S1	S0	
0	0	Desplazamiento a la derecha
0	1	Inhibición
1	0	Carga paralelo
1	1	Desplazamiento a la izquierda