Pràctica 2. Sistemes Digitals

Control d’un monitor VGA.

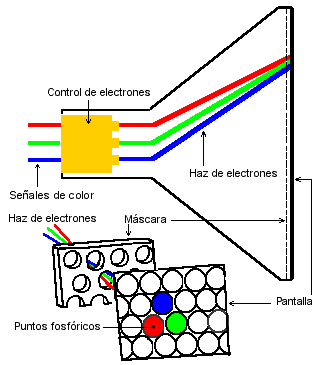
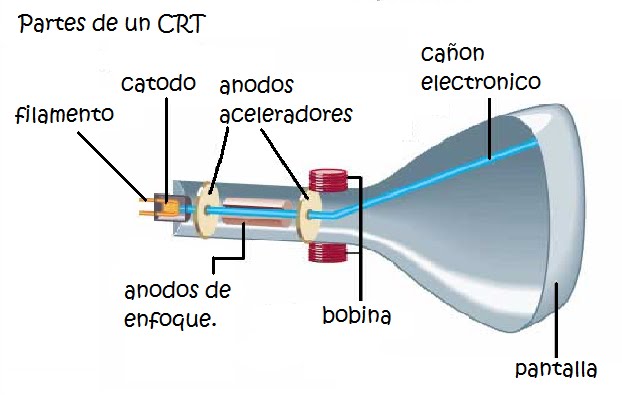
Previ 1.

El CRT o Cathode Ray Tube, va ser inventat en 1897 pero no el van utilizar fins a la creació de la televesió. Ens permet veure imatges a causa d’un feix de llum de rajos catòdics dirigits per camps elèctrics.

En els aparells de televisió i els monitors d’ordinador, la zona frontal del tub s’escaneja repetidament en un patró fix. Es controla la intensitat de cada un dels tres feixos d’electrons per a tal de producir una imatge, amb un senyal de vídeo com a referència.

Un CRT ideal hauria de ser tancat per ambdós costats amb bobines Helmholtz per a poder aplicar un camp magnètic variable i poder dirigir el feix cap a on ens interessi en aquell moment.

Com a qüestió de seguretat, la cara es fa típicament de vidre gruixut de plom per tal de ser resistent a les trencaments i bloquejar la majoria de les emissions de raigs X, sobretot si el CRT esta incorporat en un producte de consum.

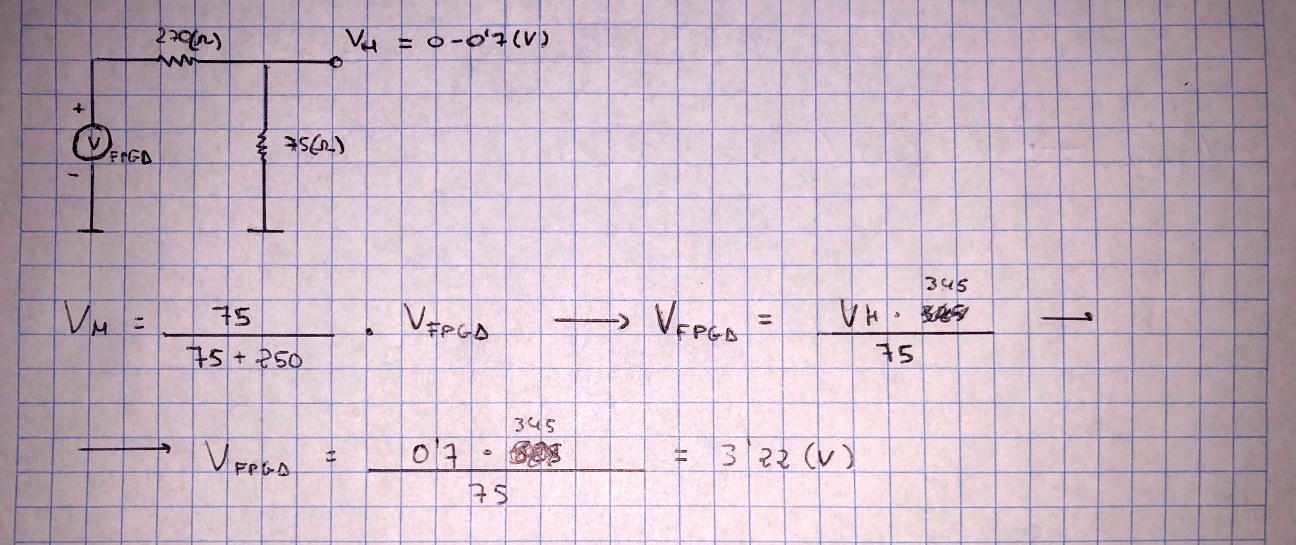


Previ 2.

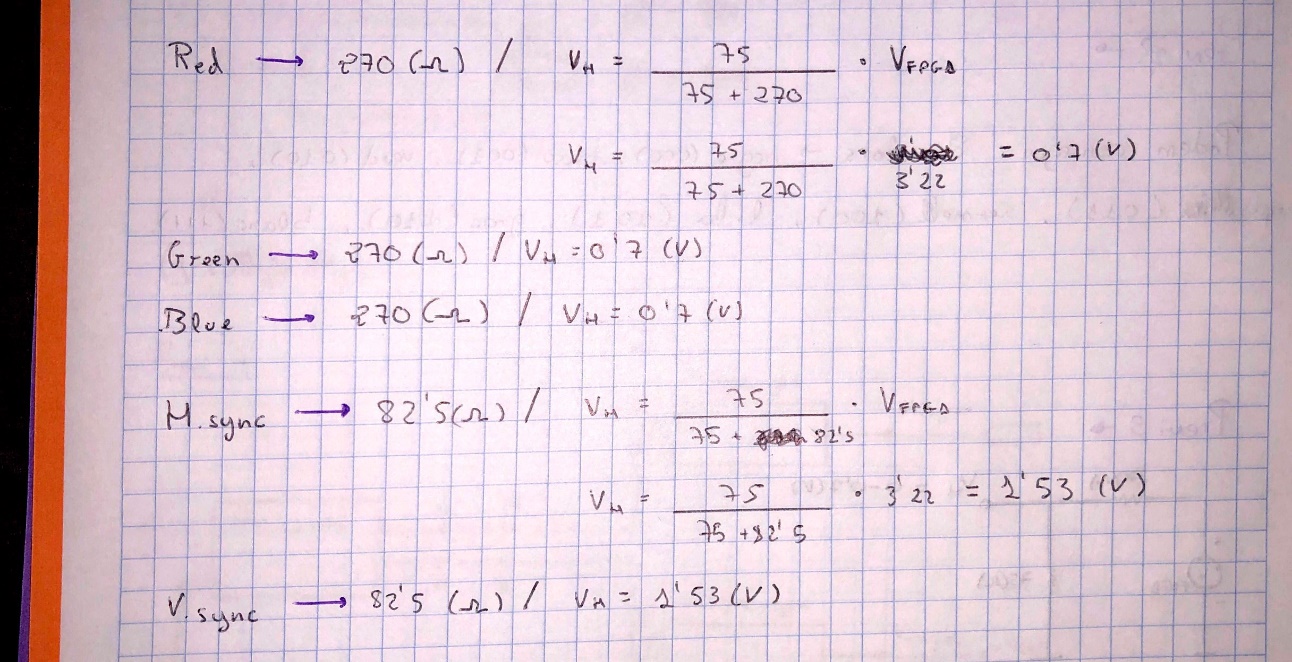
Podem aconseguir 8 colors:

Negre(000) - Blau(001) - Verd(010) - Turquesa(011) - Vermell(100) - Lila(101) -Groc(110) - Blanc(111)

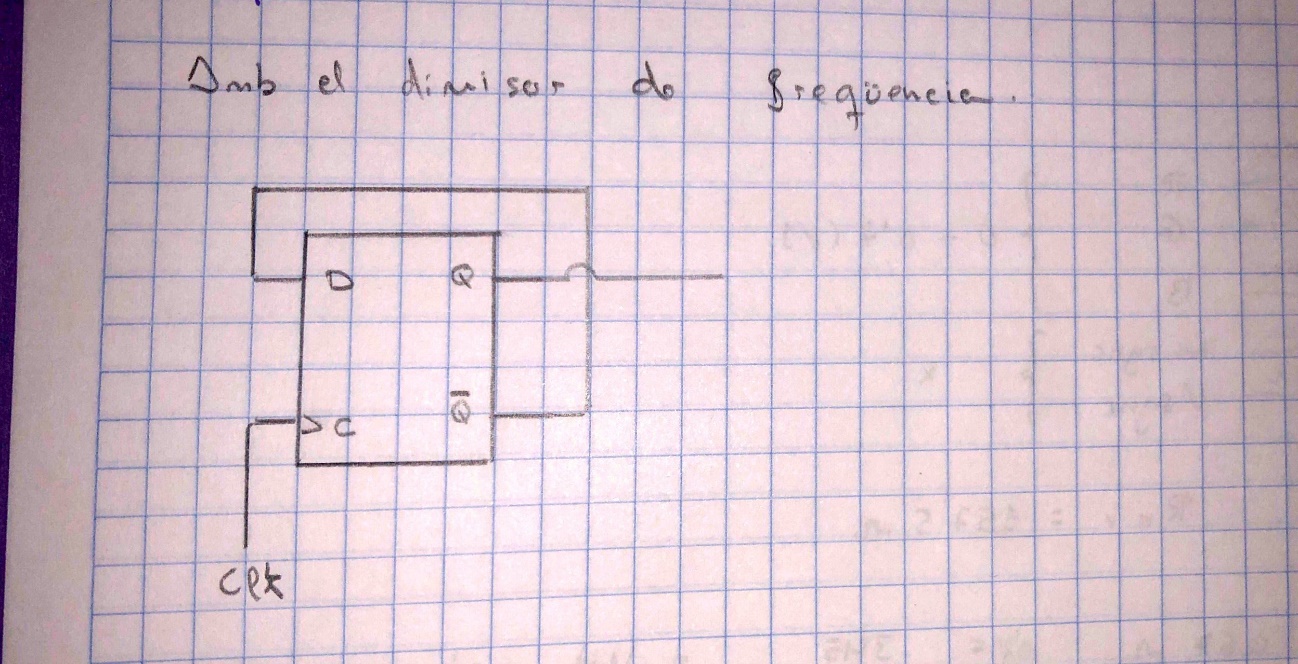
Previ 3.



Previ 4.



Previ 5.



Previ 6.

library IEEE;

use IEEE.std\_logic\_1164.all;

entity HVcounter is

Port ( clk : in STD\_LOGIC ;

H : out STD\_LOGIC\_VECTOR (9 downto 0);

V : out STD\_LOGIC\_VECTOR (9 downto 0));

end HVcounter ;

architecture arch of HVcounter is

signal h: unsigned(9 downto 0) := "0000000000"

signal v: unsigned(9 downto 9) := "0000000000"

begin

process(clk) is

begin

if rising\_edge(clk) then

if h="799" then

if v="524" then

H<=h;

V<=v;

else

h<='0';

v<=v+1;

else

h<=h+1;

end if;

end if;

end process;

end arch;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_1164.all;

entity testbench is

end testbench;

architecture tb of testbench is

component HVcounter is

port(

clk : in STD\_LOGIC ;

H : out STD\_LOGIC\_VECTOR (9 downto 0);

V : out STD\_LOGIC\_VECTOR (9 downto 0));

end component;

signal clk:

signal h: unsigned(9 downto 0) := "0000000000"

signal v: unsigned(9 downto 9) := "0000000000"

signal H: std\_logic\_vector( 9 downto 0);

signal V: std\_logic\_vector ( 9 downto 0);

constant clc\_period: time:= us;

begin

Previ 7.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_1164.all;

entity H\_sync\_gen is

Port (

H : in STD\_LOGIC\_VECTOR (9 downto 0);

H\_sync : out STD\_LOGIC);

end H\_sync\_gen;

architecture arch of H\_sync\_gen is

begin

process(H)is

begin

if H<"97" then

H\_sync<='L'

elsif H>"96" then

H\_sync<='H'

end if;

end process;

end arch;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_1164.all;

entity V\_sync\_gen is

Port (

V : in STD\_LOGIC\_VECTOR (9 downto 0);

V\_sync : out STD\_LOGIC);

end V\_sync\_gen;

architecture arch of H\_sync\_gen is

signal aux: unsigned(1 downto 0):= "00";

signal aux2: unsigned( 1 downto 0):="00";

begin

process(V)is

begin

if V<"3" then

V\_sync<='L'

elsif V>"2" then

V\_sync<='H'

end if;

end process;

end arch;

Previ 8.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_1164.all;

entity RGBgenerator is

Port (

H : in STD\_LOGIC\_VECTOR (9 downto 0);

V : in STD\_LOGIC\_VECTOR (9 downto 0);

R : out STD\_LOGIC;

G : out STD\_LOGIC;

B : out STD\_LOGIC);

end RGBgenerator;

architecture arch of RGBgenerator is

begin

process(H,V) is

begin

if V<"3" then

R<='0';

G<='0';

B<='0';

elsif V>"1" then

if V<"11" and V>"1" then

if H>"95":

R<='0';

G<='0';

B<='0';

end if;

elsif V>"11" then

if H>"95" and H<"112":

R<='0';

G<='0';

B<='0';

elsif H>"111" and H<"752" then

R<='1';

G<='0';

B<='0';

elsif H>"751" then

R<='0';

G<='0';

B<='0';

end if;

end if;

end if;

end process;

end arch;