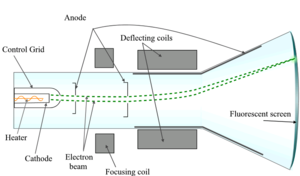
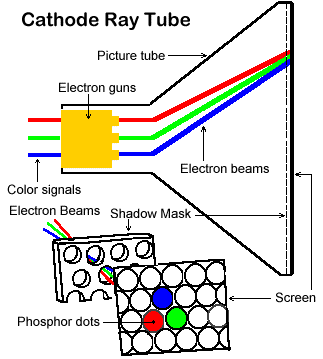
***Previ 1:***

Un CRT ("Cathode Ray Tube"), inventat el 1897 per Carl Ferdinand no va ser utilitzat fins a la creació de la televisió, és un tipus de tecnologia que ens permet veure imatges gràcies a un feix de llum de rajos catòdics dirigits per camps elèctrics.

  
Els feixos es dirigeixen cap a una pantalla de vidre recoberta de fòsfor i plom fent ús de camps elèctrics; el fòsfor emet visualitzar la imatge, mentre que el plom impedeix que la radicació creada pels feixos arribi a les persones que fan ús de la pantalla.  
Un dels efectes que té i que hem jugat amb ell quan érem petits, és l'electricitat estàtica generada al fer ús d'aquest tipus de pantalles amb la tecnologia CRT.

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; en cas de no tenir aquest tipus de bobines, un imant de neodimi suficientment fort, ens serviria, ja que ens permetria modificar la direcció del feix.

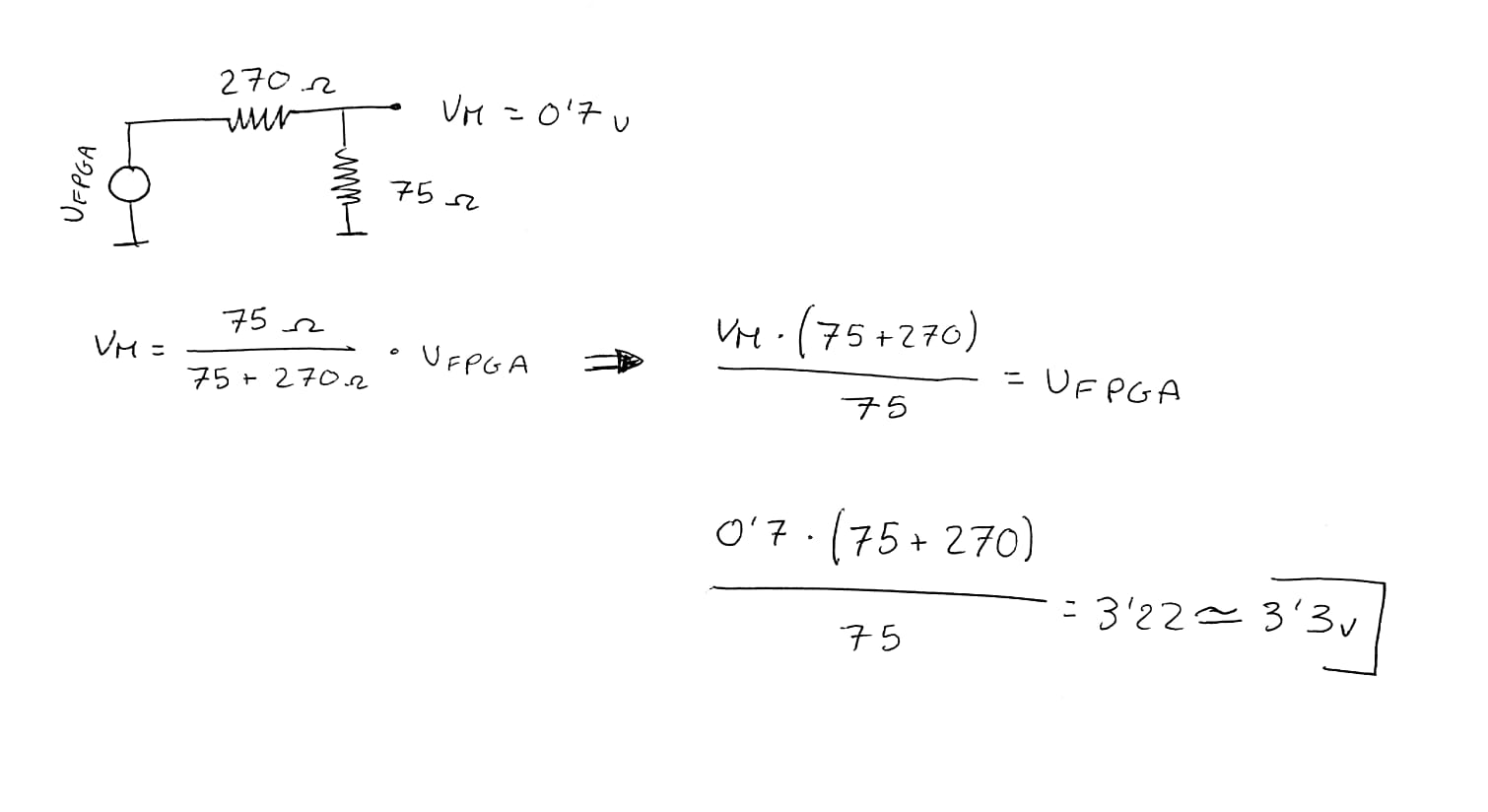
La imatge de la pantalla, depenent dels electrons que toquin el fòsfor en la part posterior, i això el que farà serà que emetin diferents colors de llum. Però hi ha vegades que els electrons es veuen obligats a anar a un lloc equivocat per un mal estat d'algun dels components explicats anteriorment, causarà una distorsió de la imatge i els famosos colors psicodèlics.



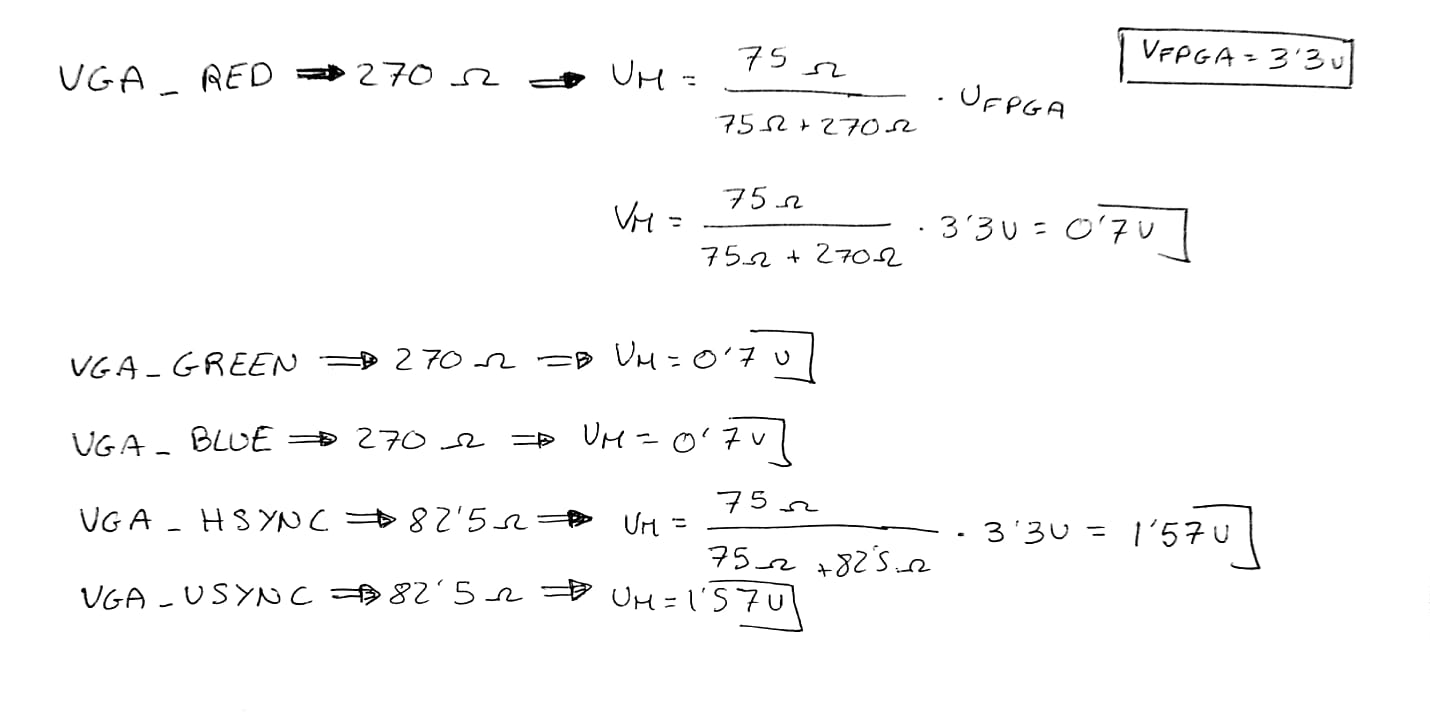
***PREVI 2:***

|  |  |  |  |
| --- | --- | --- | --- |
| **R** | **G** | **B** | **Colors** |
| 0 | 0 | 0 | Negre |
| 0 | 0 | 1 | Blau |
| 0 | 1 | 0 | Verd |
| 0 | 1 | 1 | Turquesa |
| 1 | 0 | 0 | Vermell |
| 1 | 0 | 1 | Lila |
| 1 | 1 | 0 | Groc |
| 1 | 1 | 1 | Blanc |

***Previ 3:***

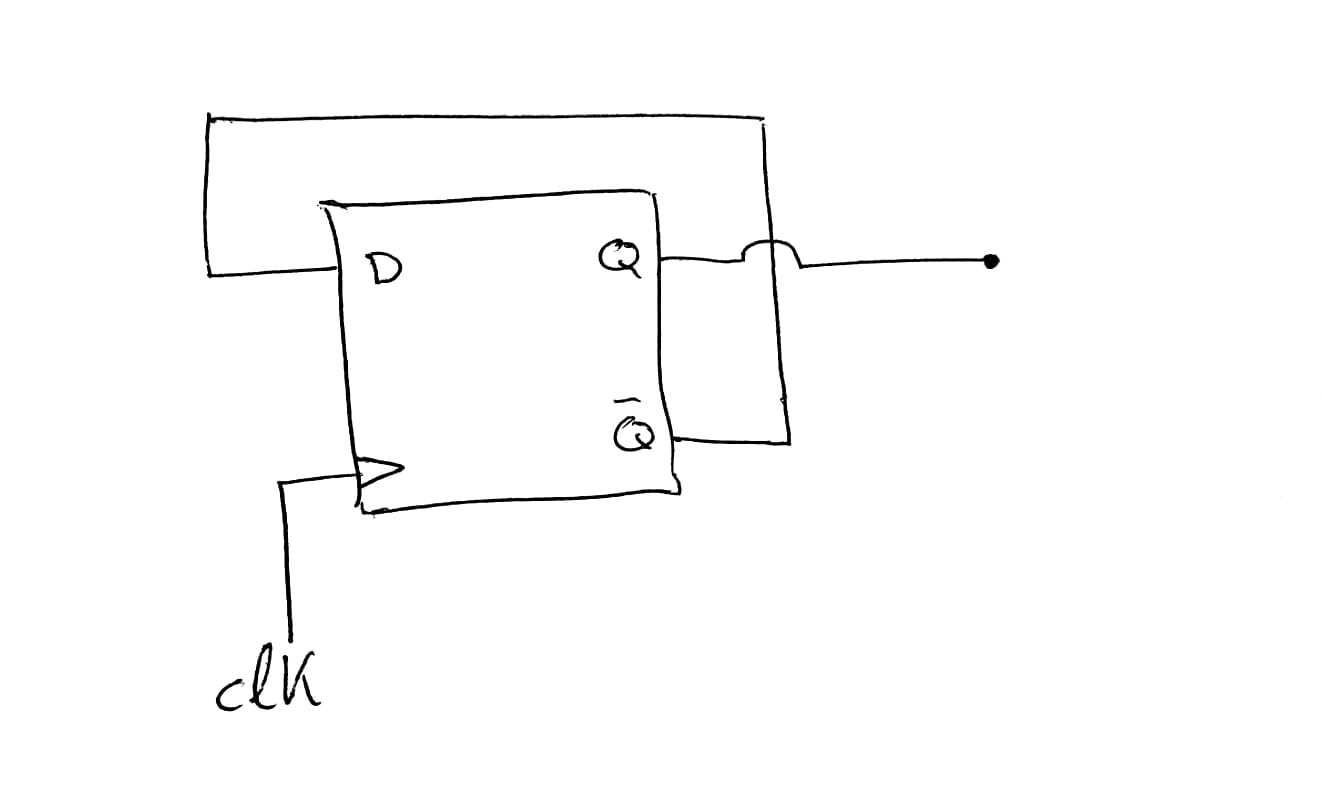


***Previ 4:***



***Previ 5:***

Amb un divisor de freqüència.



***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;

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

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;