



# Pràctica 2. Sistemes Digitals

## Control elemental d'un monitor VGA.

Pere Palà Schönwälder

Març de 2025

En aquesta pràctica generarem els senyals de sincronisme necessaris per a comandar un monitor VGA en la resolució de 640 x 480 píxels.

Aquesta pràctica dura dues sessions i per tant el document està dividit en dues parts. La primera consisteix en realitzar la simulació dels mòduls que s'han creat, i en la segona s'implementarà tot el disseny per controlar un monitor VGA.

Per a la implementació, farem servir un entorn de desenvolupament basat en programari lliure, a partir de Yosys i nexprn.

Recordeu de fer l'estudi previ! Això és imprescindible per poder accedir al laboratori.

## 1 Part 1: Simulació

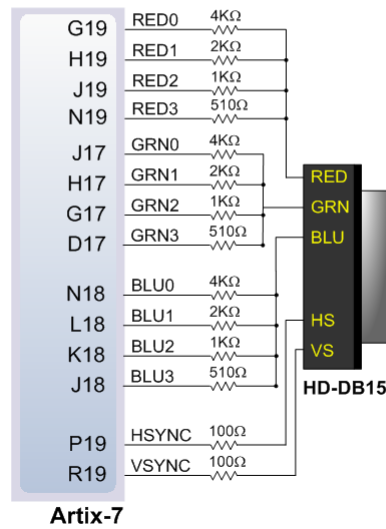
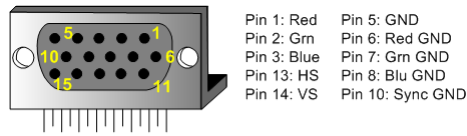
### 1.1 Els senyals VGA

Els monitors VGA estan dotats d'un connector de 15 pins que transporta diversos senyals. Els fonamentals són

- Sincronisme horitzontal: `VGA_HSYNC`
- Sincronisme vertical: `VGA_VSYNC`
- Senyal vermell (*R*): `VGA_RED`
- Senyal verd (*G*): `VGA_GREEN`
- Senyal blau (*B*): `VGA_BLUE`

més els 5 retorns corresponents a cadascun d'ells: `GND`.

A la placa de desenvolupament que fem servir al laboratori, una Basys 3 de Digilent, construïda al voltant d'una FPGA de Xilinx (model XC7A35T-1CPG236C), hi ha, entre altres elements, un connector pensat per ser connectat a un monitor analògic segons l'estàndard VGA. Aquest connector té l'aspecte que es mostra a la figura següent que també mostra com està connectat cada pin del connector amb els pins de la FPGA.



Un monitor *monocrom* simple es construeix al voltant d'un tub de raigs catòdics (TRC o CRT en anglès). En un CRT es genera un feix estret d'electrons que impacten a gran velocitat sobre una pantalla de fòsfor. L'impacte dels electrons produeix un punt de llum visible al centre de la pantalla.

Aplicant camps elèctrics o magnètics és possible desviar la trajectòria del feix d'electrons de forma que el punt es desplaci en una direcció o altra. Si aquests camps es controlen electrònicament és possible dibuixar línies (com fa un oscil·loscopi analògic dels que s'empren al laboratori de *Teoria de Circuits*) i superfícies senceres gràcies a la persistència de les imatges a la retina.

Per entendre el que fa un monitor en *color* us demanem que cerqueu informació pel vostre compte:

*Previ 1.* Cerqueu informació sobre el funcionament d'un CRT en color i feu-ne un resum amb les vostres pròpies paraules (no les d'una IA i res de copy & paste!). L'extensió orientativa és d'una cara A4 incloent una o dues figures.

Per tal d'assegurar la compatibilitat entre monitors i targes de vídeo s'han establert alguns estàndards per als senyals analògics presents als connectors. Així, la majoria de monitors suporten un mode bàsic de 640 x 480 píxels de resolució i una freqüència de refresc de 60 Hz. Els monitors LCD o TFT actuals, malgrat ser digitals, mantenen la compatibilitat amb aquests estàndards.

Per exemple, segons l'estàndard, el rang de nivells de tensió per als senyals *R*, *G* i *B* és d'entre 0 i 0.7 V. Si  $R = G = B = 0$  V obtenim l'absència de llum (negre) i si  $R = G = B = 0.7$  V obtenim blanc.

*Previ 2.* Tenint en compte el hardware disponible (mireu la figura anterior), quants i quins colors diferents podrem aconseguir amb el hardware disponible, tenint en compte que la FPGA dona senyals digitals?

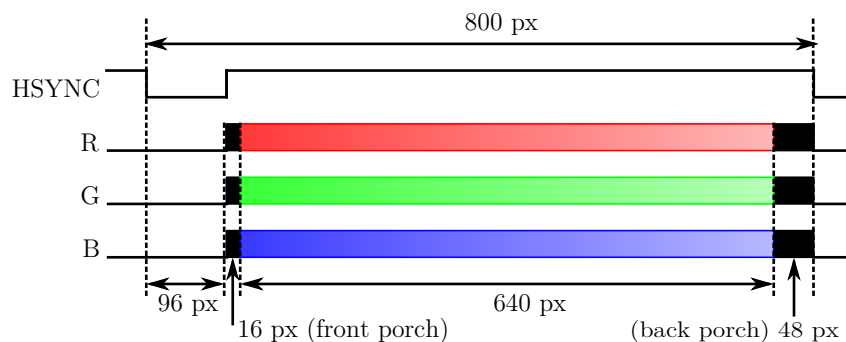
*Previ 3.* Fa una estona hem esmentat les tensions que calen al monitor. Ara afegim una informació més: el monitor incorpora 5 resistències de  $75 \Omega$  entre els 5 senyals esmentats i el node de referència GND. En assignatures posteriors esbrinarem per què s'hi incorporen. Quina tensió haurien de generar els pins de la FPGA per assegurar que als terminals *R*, *G* i *B* hi hagi els valors especificats per l'estàndar? Pista: dibuixeu acuradament el circuit equivalent format per a) el pin de sortida de la FPGA, que es comporta com una font de tensió gairebé ideal, b) la resistència que s'indica a la figura i c) la resistència que incorpora el monitor. Quines són les tensions que pot prendre el senyal RED en funció dels senyals RED3...RED0?

*Previ 4.* Assumint que la FPGA genera senyals de 0 V o de 3.3 V, calculeu quins són els nivells dels senyals de sincronisme que arriben al monitor, tenint en compte (totes) les resistències que hi ha implicades.

### 1.1.1 Estructura horitzontal (línia)

Segons l'estàndar VGA, en el mode 640 x 480 a 60 Hz, cada píxel dura  $1/25.175 \mu\text{s}$  i cada línia està formada per 800 píxels, dels quals n'hi ha 640 de visibles. La resta estan pensats perquè el feix d'electrons pugui retrocedir i establir-se adequadament.

Específicament, els cronogrames dels senyals implicats en el dibuix d'una línia són els següents:

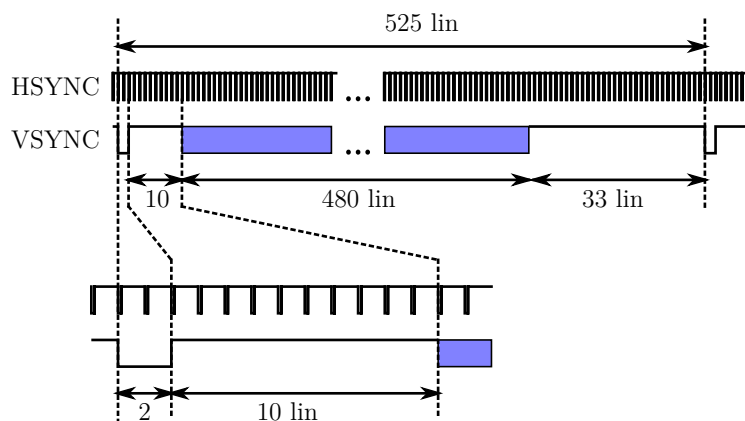


Així, el monitor interpreta que cada línia comença quan es veu un senyal baix durant el temps equivalent a 96 píxels. Després d'una pausa de 16 píxels addicionals, els 640 píxels següents s'interpreten com la informació de la línia en qüestió, començant per l'esquerra i avançant cap a la dreta. Els 48 píxels restants s'ignoren. Els senyals R, G i B haurien de ser 0 fora dels 640 píxels previstos per tal que la imatge visualitzada tingui un marc negre a esquerra i dreta.

### 1.1.2 Estructura vertical (quadre)

Per tal de saber quina línia és la primera (la superior), es genera un senyal de sincronisme vertical que pren valor baix únicament durant el temps corresponent a les dues primeres línies. Les 10 línies següents no són útils per al dibuix (durant aquestes línies els senyals R, G, B haurien de ser zero per donar una franja fosca a la part superior de la imatge).

A continuació hi ha les 480 línies útils i, per acabar, 33 línies que també s'haurien de mostrar negres. La figura següent ho mostra esquemàticament:



## 1.2 Generació dels senyals de sincronisme en VHDL

A la placa de desenvolupament es disposa d'un senyal de rellotge de 100 MHz. D'aquí serà possible generar senyals de 25 MHz que podrem fer servir com a rellotge de píxel. Malgrat que la seva freqüència hauria de ser 25.175 MHz, la majoria de monitors són capaços de sincronitzar-se amb freqüències de rellotge lleugerament diferents.

El senyal de 25 MHz es pot generar a partir del clock principal de 100 MHz a partir d'uns blocs anomenats PLL. No obstant, són blocs relativament complexes i tenen un suport limitat amb el toolchain que farem servir. En el cas que ens ocupa podem trobar opcions molt més simples.

*Previ 5.* Escriviu un codi VHDL que permeti generar un senyal de **enable** que s'activa durant un període de clock i ho fa a una freqüència de 25 MHz. Disposeu d'un rellotge de 100 Mhz.

Com a base per fer la generació de sincronismes es proposa que desenvolueu un mòdul VHDL amb dos comptadors: un comptador horitzontal, que va comptant de 0 a 799 i un comptador vertical, que compta de 0 a 524 cada vegada que l'anterior es reinicialitza a 0. Els ports que hauria de tenir són:

```
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;
```

*Previ 6.* Descriviu un bloc en VHDL que implementi els dos comptadors segons el template anterior. Descriviu, també, el fitxer de test per a la simulació.

A partir d'aquests comptadors podem generar els sincronismes horitzontal i vertical, activant o desactivant un senyal en funció del valor dels comptadors corresponents. Per exemple, podríem pensar en un mòdul descrit per:

```
entity H_sync_gen is
  Port ( H       : in    STD_LOGIC_VECTOR (9 downto 0);
        H_sync  : out   STD_LOGIC);
end H_sync_gen;
```

per generar els sincronisme horitzontal i un de similar per al vertical.

*Previ 7.* Doneu un codi VHDL per als blocs `H_sync_gen` i `V_sync_gen`. Desenvolpeu un testbench que permeti verificar el correcte funcionament dels senyals de sincronisme.

A continuació, es tractarà de pintar la pantalla útil (els píxels adequats) de color blau.

*Previ 8.* Escriviu el codi VHDL del bloc `RGBgenerator` de forma que els senyals R, G i B siguin els adequats per omplir la zona útil de la pantalla de color blau.

```
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;
```

*Tasca 1.* Simuleu els vostres blocs corresponents al *Previ 6*, *Previ 7* i *Previ 8*. Adjunteu les gràfiques en el informe, en format pdf, que mostrin el seu correcte funcionament. Recordeu que si heu fet canvis en el codi VHDL respecte als previs també l'heu d'adjuntar en el informe.

## 2 Part 2: Implementació

Després de la primera sessió on s'han comprovat els blocs en VHDL simulant, només falta implementar-ho en el laboratori.

*Previ 9.* Modifiqueu el bloc `RGBgenerator` per omplir la pantalla amb tres franges verticals. La de més a l'esquerra de color blau, la del mig de color verd i la de la dreta de color vermell.

*Tasca 2.* Feu un disseny on el document de màxim nivell sigui un esquemàtic. Implementeu els blocs VHDL necessaris per generar els polsos de sincronisme, segons el que heu fet a l'estudi previ i interconnecteu-los adequadament.

*Tasca 3.* Comproveu amb l'oscil·loscopi que els senyals de sincronisme són els adequats. Específicament, comproveu la seva freqüència, la durada dels polsos i la seva posició relativa. Feu les captures corresponents per adjuntar a l'informe.

*Tasca 4.* Implementeu la generació dels senyals R, G i B per aconseguir que tota la pantalla sigui verda al vostre disseny VHDL. Comproveu amb l'oscil·loscopi que el senyal G sols s'activa en els moments adequats. Feu les captures corresponents per adjuntar a l'informe.

*Tasca 5.* Connecteu un monitor VGA a la placa i comproveu que la visualització és correcta.

*Tasca 6.* Modifiqueu el codi del bloc `RGBgenerator` per dividir la pantalla en tres franges horitzontals de tres colors diferents.