

LOGO

Università degli Studi di Genova

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
CORSO DI LAUREA MAGISTRALE IN FISICA

Controllo di uno schermo tramite porta VGA
attraverso un FPGA

Tesina per Elettronica Applicata

Alex Amato

Daniele Rapetti

Sommario

Abstract

Anno 2014-2015

Indice

1		3
1.1	Preliminari	3
1.2	Un frame	3

1

Per controllare un display con una vga serve sapere delle cose

1.1 Preliminari

Bisogna conoscere il framerate dello schermo, ovvero la quantità di volte al secondo in cui viene disegnata l'immagine su di esso. Per ogni frame visualizzato lo schermo per un certo periodo rimane nero, bisogna tenere conto di questo vuoto con un parametro detto **retrace factor**. Infine bisogna aver presente la risoluzione dello schermo (ad esempio 1280×1024).

Con questi parametri si calcola il pixelclock, ovvero la frequenza con cui vengono mandate le informazioni allo schermo:

$$PixelClock = \frac{(Horiz\ Res) \times (Vert\ Res) \times (Frame\ Rate)}{RetraceFactor} \quad (1)$$

1.2 Un frame

Ora cercherò di descrivere come funziona un frame, poi Alex leggerà e correggerà.

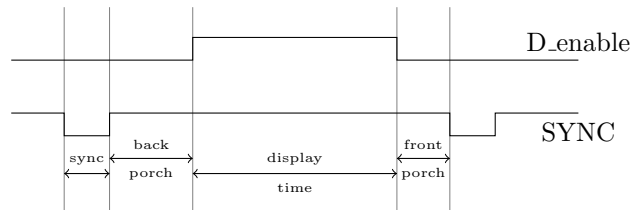


Figura 1: I segnali sincronismo e display

Per gestire le tempistiche del display devo rispettare delle precise indicazioni:

Ho un segnale principale che si chiama *HSYNC* che indica allo schermo quando cambiare riga, è normalmente alto. Dopo il segnale di *HSYNC* devo aspettare un tempo chiamato **back porch** dopo il quale c'è il *display time* che in impulsi di clock, coincide con il numero di pixel sulla linea, durante il quale mi occupo di inviare allo schermo le informazioni sul colore del singolo pixel, quindi c'è un ulteriore tempo morto, detto **front porch** in cui abbasso l'enable del display e aspetto il seguente segnale di sincronia.

Per il segnale di sincronia verticale il discorso è lo stesso, con la differenza che le durate dei vari segmenti sono "linee orizzontali" invece che impulsi di clock.

Per controllare i quattro segnali, dato che hanno la stessa struttura due a due abbiamo deciso di realizzare due macchine a stati sincrone uguali, con dei parametri variabili:

```
module stm_timing(  
  
    input  clk          ,  
    input  rst_n        ,  
  
    output o_sync      ,  
    output o_disp      ,  
  
);  
  
//=====   
// PARAMETRI   
//=====   
  
parameter Disp = 1280 ;
```

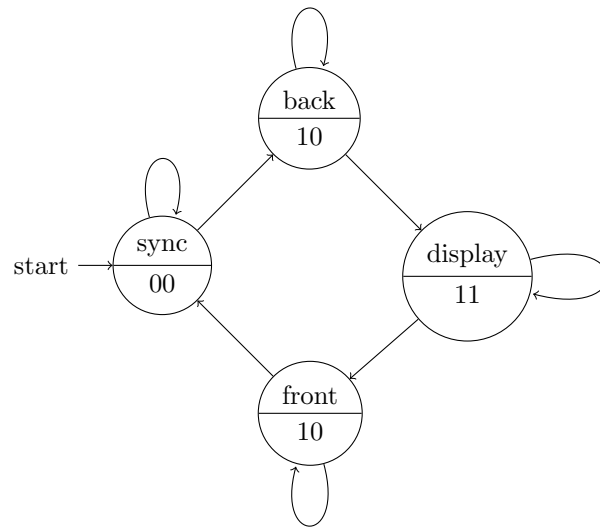


Figura 2: Il funzionamento della macchina a stati spiegheremo in seguito le frecce, quando Alex farà la revisione

parameter	Front	= 48	;
parameter	Sync	= 112	;
parameter	Back	= 248	;