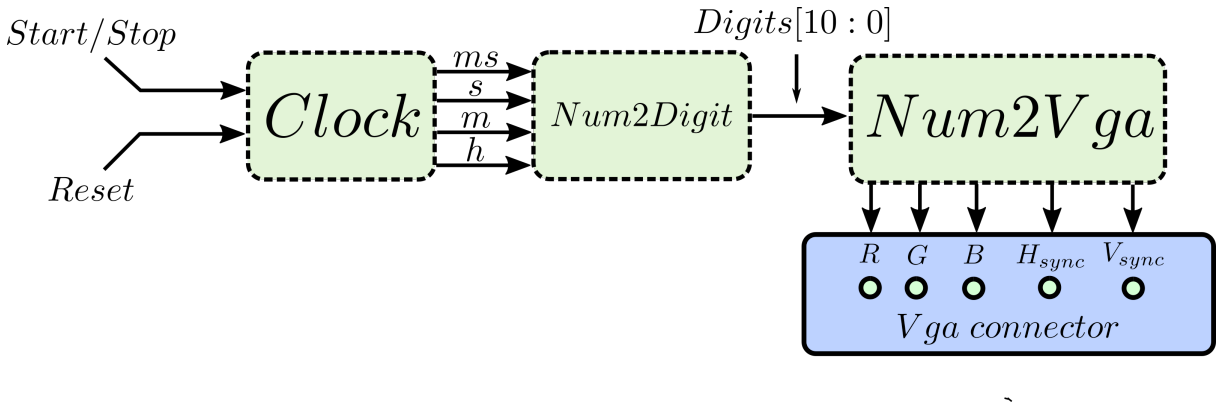
INSTITUTO TECNOLÓGICO DE BUENOS AIRES

Trabajo Práctico Final

NOWIK, Ariel 58309
MASPERO, Martina 57120
MESTANZA, Joaquin 58288
REGUEIRA, Marcelo 58300
November 12, 2018

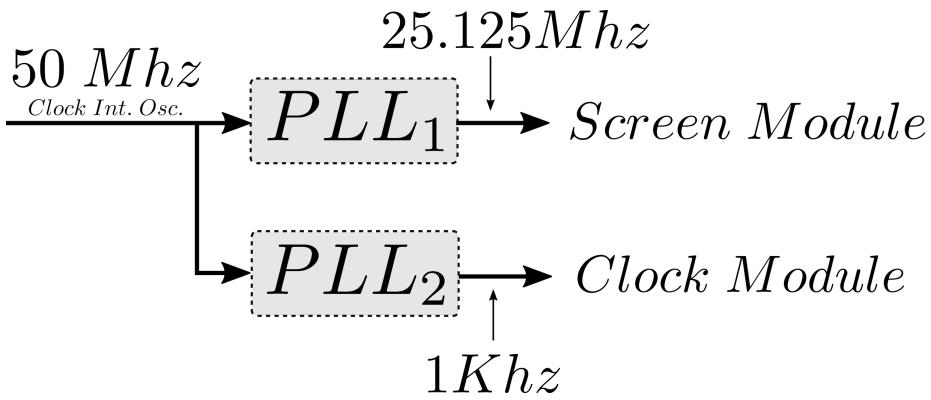
Module structure



The project is organized in three modules:

1. Clock module: handles the clock reset/start/stop actions (via Buttons), and counts milisecconds, seconds, minutes and hours.
2. Number to digit module: Receives the numbers from the clock and converts them to digit codes.
3. Digit to Screen module: Gets the numbers and handles the Vga protocol.

Clock syncronization



The project uses FPGA internal 50Mhz oscillator. Using two PLLs, the different frequencies needed for each module is generated

Clock Module

This module recives button input and organizes the counting of the chronometer. Its output are the numbers of miliseconds, seconds, minutes and hours. It's implementation is very simple and it is not worth explaining it in detail.

Number to Digit Module

This module recives the number inputs and converts them to digit codes. That digit code information represent which characetrers will be shown in screen. As there are also ':' character on screen, also there is a digit code 10, that represent that character. There are, in total 10 digits on screen.

Digit to Screen Module

This is the most advanced, and difficult to debug module. The objective, however is simple: Show the digits communicated by the digit module on screen one after the other. When we developed this module we've got a problem. It was not possible, at least directly, to store all 640x800 bits on memory and then access to them, write them, or pass them from one module to another. So we worked hard to find a solution that didn't need that. The idea is to, in place, when each pixel is drawn, decide, if it will be drawn, or not. We tried to use a simple formula

The magic Lines