

# LED DISPLAY COMMAND SYSTEM USING ZYNQ 7000 SYSTEM ON CHIP

Laurentiu – Mihai IONESCU<sup>1</sup>, Alin Gheorghită MAZĂRE<sup>1</sup>, Nadia BELU<sup>1</sup>, Marius CÎRSTEA<sup>2</sup>

<sup>1</sup>University of Pitesti, Romania

<sup>2</sup>Visionware SRL., Bucharest, Romania

<sup>1</sup>laurentiu.ionescu@upit.ro

Keywords: System on Chip (SoC), LED display, web server application

*Abstract: In this paper we present a complete solution of command system for LED display using a system on chip architecture. Our solution involves the hardware multiplexer for columns synchronization, the parallel processing module for image preparation and the web software interface for image file selection and conversion in a compatible format. While the hardware processing modules are implemented on FPGA module (Field Programmable Gates Array), the web interface and the web server run on processor (ARM architecture). Both FPGA and processor are part of a single chip system manufactured by Xilinx. Thus are combined the flexibility provided by software part which run on processor and the fast processing speed provided by hardware parallel modules. The paper presents the result obtained with our solution which commands a LED table with 160x48 resolutions with size of 2 m x 0.5 m.*

## 1. INTRODUCTION

Currently there are commercial solutions for displaying information on LED screens [1]. However, many of these solutions have certain limitations, especially the necessity of a computer for image processing and low speed in image displaying.

This paper describes the design of a control system for a table display with LED. The system must to meet several conditions imposed by its practical use. First system must manage a large display table with high number of LEDs. For example, a table with 160x48 resolution as the one that will be used in our project will be administered a total of 7,680 LEDs. If they are grouped into 8 columns then we have a number of 960 LEDs to be configured at a time. By using the multiplexing technique, a requirement that follows from this is to set up a large number of LEDs (960) in a time interval corresponding to the frequency of at least 400Hz. Exceeding it, so work at lower frequencies per frame, would have the effect of altering the quality of the displayed

image. The use of a microprocessor system could not provide sufficiently high calculation speed due to the fact that all the operations are performed sequentially. This is why we opted for a structure of reconfigurable FPGA hardware where can be designed modules which run parallel tasks. Display tasks were assigned for each module which makes its operation independently of other. The separation between the modules is via the bi-port buffers memories. On the other hand, the information displayed, which consists of static and dynamic images, must be changed in a simple manner, without big interventions in the internal structure. At the same time, another goal is to have a high degree of autonomy and to be more integrated. For these reasons, the solution using a computer to transmit images and montages of images is not acceptable. Instead we opted to use an embedded processor in a System on Chip (SoC) device. Xilin Zynq 7000 device contains an embedded processor and a FPGA module integrated on single chip. It allows writing assemblies pictures and images easily.

In the next section will show the structure of the system, then the command structure implemented PL (Programmable Logic - FPGA) side of SoC and the algorithm and web application designed to running on the PS (Programmable Structure - processor) side of SoC.

## 2. SYSTEM COMPONENTS

The figure below presents the main component of the system.

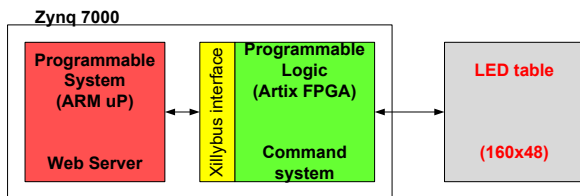


Fig.1 System block diagram

The control system is integrated into a System on Chip (SoC) from Xilinx. SoC contains two components: an ARM processor (called Programmable System - PS) and a FPGA Artix (Programmable Logic - PL). The processor is responsible for running a server that allows users to upload images and convert them in format supportable by LED table. On the PL side has developed a system that controls the LED table. Communication PS - PL is performed using an interface based on the bi-port SRAM memory. On a port PS write images processed and on the other port PL retrieves images and transmits them to the table with LEDs.

## 3. COMMAND SYSTEM

The figure below presents a diagram of the control system designed in PL.

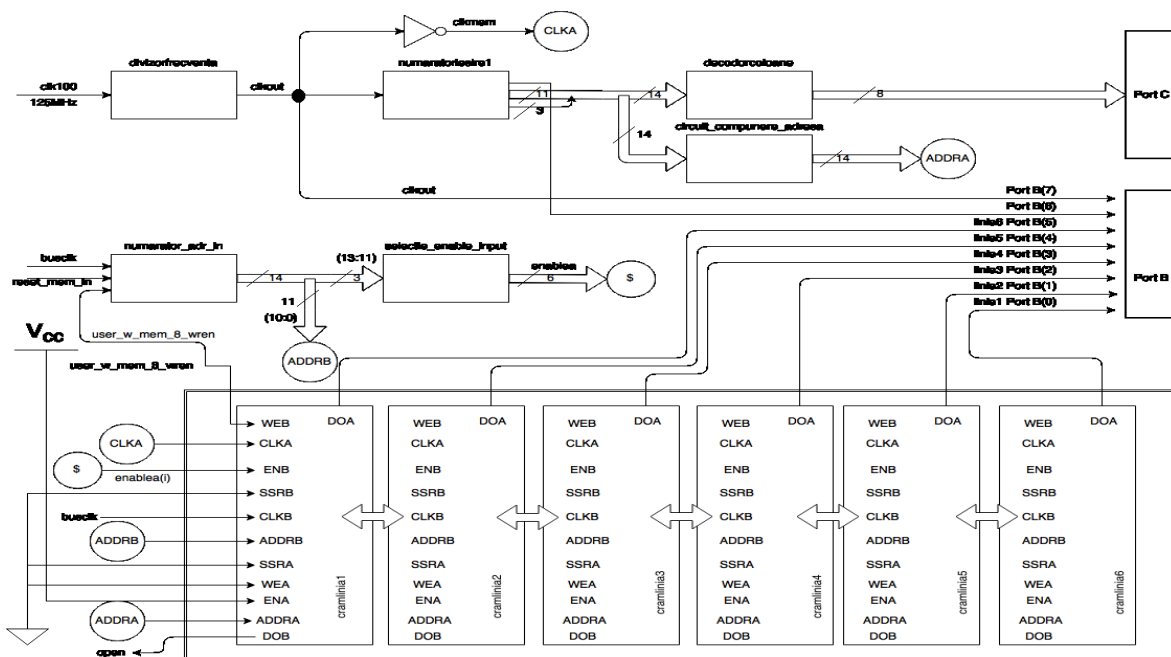


Fig.2 Command system implemented in PL side of SoC - diagram

As can be seen in Figure 2, we used six bi-port memories, such S1\_S9 (organization 16k x 1 port A, 2k x 8 + 1 port B), two clock signals (busclk, clk100) first provided from logic XillyBus interface and the second with fixed frequency of 100MHz, a reset signal which is controlled by XillyBus too, and user\_w\_write\_8\_wren signal, which is activated when writing files to perform a specific XillyBus interface operation with system memory (eg / dev / xillybus\_mem\_8 [3]).

Besides taking signals from the interface, we designed addressing circuits (counters, decoders), and a more complex circuit for composition of the index memory area.

Output ports B and C will contain the necessary data table to display information stored on the PL side via application server which run on PS.

Thus, the port C is responsible for sending commands for LEDs cathodes (from K1 to K8), common for clusters. Port B is responsible for sending data lines using clock signals generated

on the same B port (‘ST’ – port B(6) – load serial bit, ‘SH’ – port B(7) – load parallel bit).

#### 4. WEB SERVER APPLICATION

The software part of the project is the application web server implemented in Xillinux operating system (run on PS from Zynq 7000 SoC) using XAMPP package. The package is composed of PHP language compiler and MySQL as a system management database [2].

The database contains information on the application users. To interact with LEDs table an user must have an account. Thus, changing the image of the LED table is possible only after logging in to the site.

Software consists of three major parts:

- logging and registration,
- image file upload and conversion,
- image upload to table.

##### 4.1. Login and registration

First, the platform required to access a user account. In figure 3, left, the image of the login screen is observed. It shows two text fields where the user will enter his credentials.



Fig.3 Login and register web interfaces

If a user has no account, it will need to create one by pressing the record link.

In figure 3, right, is a capture with register form with fields to be filled to create an account to the platform. The fields marked with an asterisk are mandatory. So to have an account in the platform takes only a username and password are required.

All the user’s details will be saved in a database. As management system we used MySQL database version 5.5.

##### 4.2. Main page

In the figure 4 you can see the main screen of the application. This contains two buttons, and a link to log (end of work session).

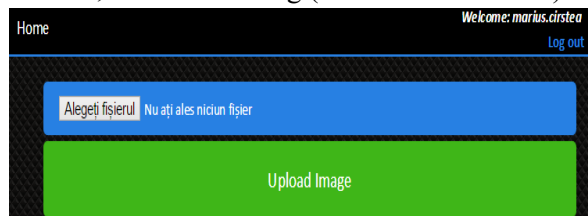


Fig.4a) Main page without any image file selected

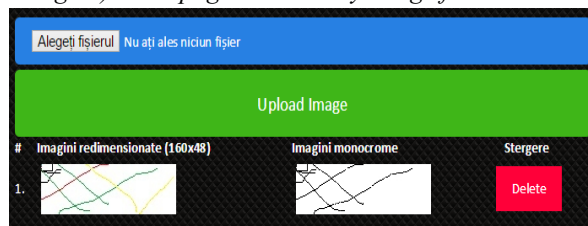


Fig.4b) Main page with an image selected

After pressing the upload, you can see how the selected image was uploaded, and also processed.

##### 4.3. Image preparation

As can be seen in figure 5, after uploading an image file will be created 3 new files on the server:

- The first file is the original, unmodified image,
- The second file is image resized to a width of 160 pixels and a height of 48 pixels,

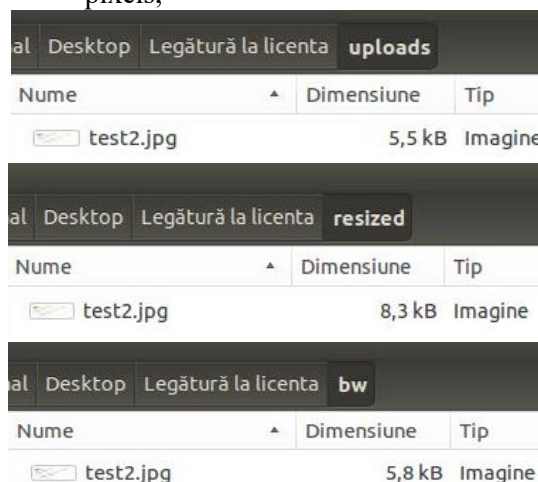


Fig. 5 Three files representing the three steps in image preparation: original file uploaded (top), image resized (middle) and bw conversion (bottom)

- The third file will be the image converted to monochrome using the previously processed file (resized image).

### 5. TESTING THE SYSTEM

Our system was designed using a Digilent Zybo board: the board is based on Xilinx Zynq 7000 SoC. To build a workstation with Zybo board are necessary the following:

- A monitor capable of 1024x768 display resolution with VGA port,
- An VGA monitor cable,
- An USB keyboard,
- An USB mouse,
- A USB hub compatible with Linux 3.12.0.



Fig. 6 Workstation with Zybo board used in tests (Zybo is based on Zynq 7000 SoC)

Zybo board supports also digital connection via a HDMI port, which displays the same image as VGA port.



Fig.7 Image preview before upload to table

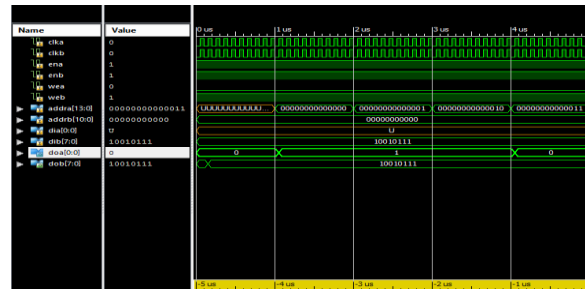


Fig.8 Print screen from Xilinx ISim with simulation of BRAM operation



Fig.9 Image with table while displaying. In the right corner of picture is Zybo workstation

### 6. CONCLUSIONS

Practically Zynq SoC in combination with LED tables brings the ability to display a message or a custom image of 2 clicks in a web interface which can be accessed from laptops, PC, tablets or smart phones.

The implementation was done using 6 bi-port memories, which led to a safe storage of the pixel values that are displayed. To improve this work, we could consider displaying to a color LED table. Also, a possible option to make the project more interesting is implementing successive frames of images to create the effect of movement.

### 7. REFERENCES

- [1]. Tehnium Azi, „LED-uri si montaje din LED-uri”, 12.04.2014, [http://www.tehnum-azi.ro/page/articole\\_articles/\\_/articles/notiuni-teoretice-din-electronica/led-uri-si-montaje-cu-led-uri-r39;](http://www.tehnum-azi.ro/page/articole_articles/_/articles/notiuni-teoretice-din-electronica/led-uri-si-montaje-cu-led-uri-r39;)
- [2]. Robert Schifreen: „How to create Web sites and applications with HTML, CSS, Javascript, PHP and MySQL”, Oakworth Business Publishing Ltd, UK, 2009.
- [3]. Xilinx, „Xillybus getting started”, 2015, [http://xillybus.com/downloads/doc/xillybus\\_getting\\_started\\_zynq.pdf](http://xillybus.com/downloads/doc/xillybus_getting_started_zynq.pdf).