National and Kapodistrian
University of Athens

Department of Informatics
and Telecommunications

Design and implementation of an Embedded System on Chip for supporting signal and image processing operations on an FPGA

Zisis M. Tsioumaras, Evangelos Logaras, Elias S. Manolakos (project PI)
{z.tsioumaras, evlog, eliasm}@di.uoa.gr

Abstract

The purpose of this project was the design and implementation of an embedded System on Chip (SoC) to support simple embedded processing of data, arrays and images. A systematic design flow has been followed, starting from setting the system's specifications, to system design and development and finally testing and validation. The hardware description language used was VHDL and the implementation was based on a Xilinx Spartan-3 FPGA, residing on the Spartan-3 development Board by Digilent Inc that is extensively used in academic institutions for educational purposes. The system is using a memory controller already designed by postgraduate students of the University of Athens, Department of Informatics and Telecommunications in a previous project. The system implements four atomic operations a user needs: READ (fetch an 8-bit word from board's SRAM), WRITE (store an 8-bit word to the board's SRAM), QUANTIZE (quantization of an 8-bit image to 3-bit) and DISPLAY (display an image through the board's VGA port). By using these operations, and based on a simple bus protocol, the user can utilize the board's external SRAM and VGA port to develop IP cores that run processing algorithms on arrays and images stored on the board's SRAM. In order to transmit data from the PC to the board's SRAM and vice versa, a library with Matlab functions was developed. The system handles data in 8-bit words. Nonetheless Matlab functions have been developed to enable the transmission of 16, 24 and 32 bit words in two's complement. Finally, representative use cases of the system are also presented.

System Units

In figure 2 we see the main system units. Memory Controller is responsible for accessing board's SRAM from a PC or an internal unit. VGA Controller displays images read from SRAM, with the assistant of the core VGA Interface, which reads image data (pixel value) from SRAM (without the use of Memory Controller) and feeds with it the VGA Controller in the correct clock cycle. In addition we developed Image Dimension Reader which reads the dimensions of the image that is to be displayed and checks for potential memory overflow. Also we developed Image Quantizer which takes a memory address and a threshold and quantizes the 8-bit image stored in this address to 3-bit, because only 3-bit images can be displayed from Sparta-3 Boards VGA port. In case of an RGB image, Image Quantizer performs apart from quantization and image compression. Control Unit receives user's IP core requests and activates the corresponding units. It is the only unit that user deals with and implements the communication protocol.

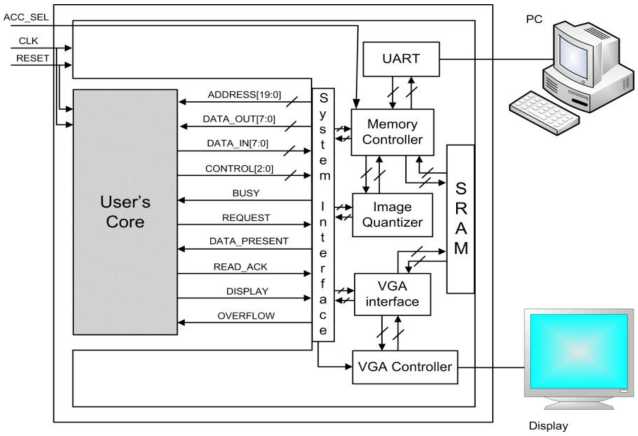

Fig 1. Top-Level system schematic. The control signals and busses along with the data exchange busses used for communication between user's component and the main system are displayed.

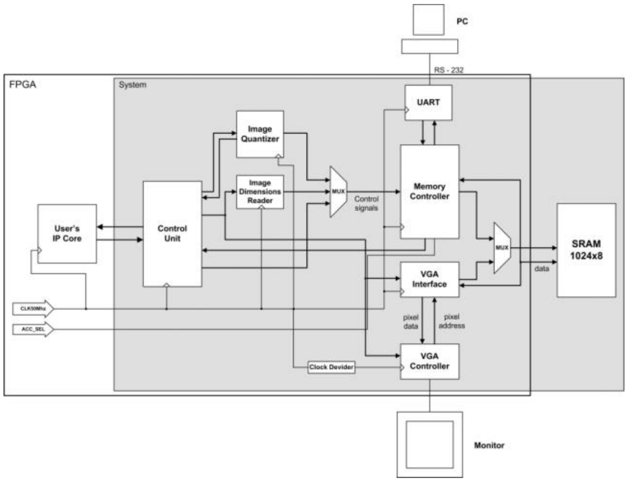

Fig 2. System Block Diagram. FPGA and System borders are displayed along with the different parts that compose the main system.

General specifications and limitations for system use

*Memory limitations***:** Spartan-3 Boards has 1MB SRAM available for data store, which limits the number of images and arrays stored at the same time in SRAM. *Monitor resolution:* VGA controller supports 640x480 pixel resolution at 60Hz refresh rate. As a result no image can be magnified beyond these dimensions. *Displaying colors:* VGA port available on Spartan-3 Board controls five signals: Red, Blue, Green, Horizontal Sync and Vertical Sync. Therefore, with one bit for each of Red, Green and Blue, eight colors are possible for display: Black, Blue, Green, Cyan, Red, Magenta, Yellow and White. *Available space on FPGA:* Spartan-3 FPGA has 4.320 equivalent logic cells, a number of which will be occupied by the main system. User should develop his IP core according to the available slices. *Data heading:* For the sound system operation, each image, vector or array stored on SRAM, should be stored with a heading that consists of three 8-bit words. In case of an image, this heading keeps information about image dimensions and (in the third word) color number. In case of an array, the number of rows and columns is kept and the third word is spare. Matlab functions also get use of data heading to calculate the 8-bit words for transmission and reception.

Communication Protocol (with user's IP core)

System provides four operations through which a user can have access to board's SRAM and to VGA interface. The four operations are: *READ* - with this operation a user can read an 8-bit word from board's SRAM, *WRITE* - this operation enables the user to store an 8-bit word to board's SRAM, *QUANTIZE* **-** with this operation a user can quantize an 8-bit image to create a 3-bit one for the latter to be displayed through board's VGA interface, *DISPLAY* – the final operation lets a user set a 3-bit image for display through board's VGA port.

Matlab Functions Library

In order to exchange data between a PC and board's SRAM, the user can use Matlab functions from a library developed for this reason, within the scope of the project. In these Matlab functions different types of data (images or arrays) are dealt in a same way, by being represented as objects with four fields. These fields are: *type* – indicates whether the data is an image or an array, *address* – field in which board's SRAM address is stored, *data* – field with the main data of the object, *precision* - variable that can be assigned the values 8, 16, 24 and 32 according to how many bits we want an array's element to be represented in (used only in arrays).

Use Cases

For system validation we developed two use cases. The first one is the implementation of an algorithm that performs image inversion (fig 4). The second one is an implementation of the Sobel operator, for image edge detection (see fig 5 for a grayscale image and fig 6 for an RGB image).

Publications

[1] Zisis M. Tsioumaras, Evangelos Logaras, Elias S. Manolakos, "Design and implementation of an Embedded System on Chip for supporting signal and image processing operations on an FPGA." In Proc. Univerisity of Athens annual volume "Selected B.Sc. and M.Sc. thesis", 2009.

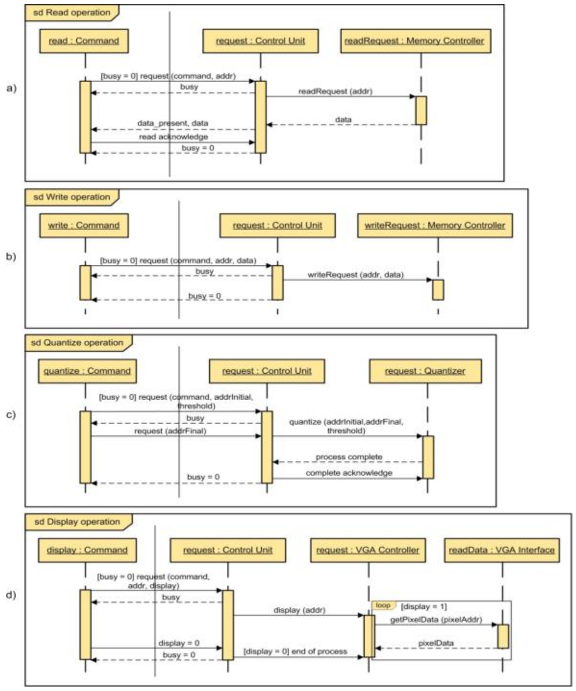

Fig 3. Operations UML Sequence Diagram

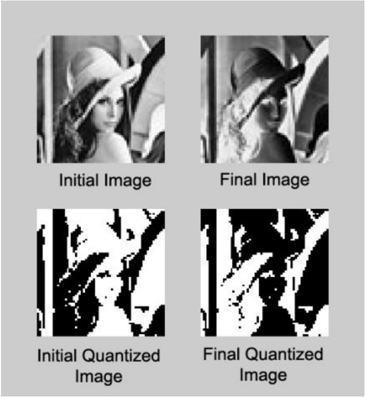

Fig 4. Results of a typical image inversion algorithm run with input a grayscale image and its quantized (black-white) version.

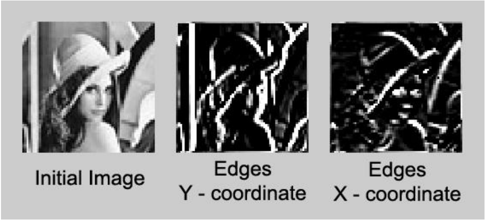

Fig 5. Edge detection using Sobel algorithm in a grayscale image

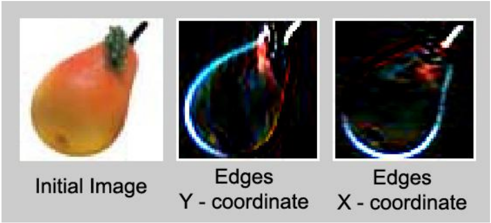

Fig 6. Edge detection using Sobel algorithm in an RGB image