ISSN: 1991-8941

Design of Embedded Image Processing System Using FPGAs

Mazen R. Khalil Sara A. Mohamed
Technical college/Mosul

Abstract: The work aims to design an image processing system to be configured on Spartan-3E FPGA. Embedded design techniques were adopted to construct a soft core processor system. Intel strata flash parallel NOR PROM is added to the system to meet the requirement of storing huge number of image samples. Suitable software driver with C language are used for purpose of erasing the flash PROM and write operations. High- pass and unsharp digital filers were used for the purpose of image processing. Matlab program is used for verifying the results.

Keywords:. Design ,Embedded Image Processing System ,FPGAs

Introduction

A high pass filter can be used for edge enhancement as it is passes only high frequency information, corresponding to places where gray level are changing rapidly The main advantage of the high pass filter is to increase the degree of sharpening. It enhance details in all directions equally.

Field Programmable Gate Arrays (FPGAs) are becoming increasingly popular for implementation of logic circuits. Inclusion of abundant memory resources in FPGAs has made them suitable for implementation of embedded systems, where a complete system fits on a single programmable chip. A processor unit in such a system is usually a soft-core processor.A soft-core processor is a microprocessor fully described in software, usually in an high description language(HDL), which can be synthesized in programmable hardware, such as FPGAs. Soft-core processors implemented in FPGAs can be easily customized to the needs of a specific target application[1].

The soft core processor solutions offered by Xilinx, which can be implemented in a Spartan-3E FPGA, are PicoBlaze and MicroBlaze. MicroBlaze processor is a 32-bit RISC softcore processor optimized for implementation in Xilinx FPGA. The implementation of this processor has been realized in the Xilinx Embedded Development Kit (EDK) design environment offered by the Xilinx company. The most important tool of this environment is the Xilinx Platform Studio (XPS) with which we can design the entire hardware structure[2].

The aim of the research is to cover stages of designing an embedded image processing system, using facilities provided by the FPGA's and EDK tools dedicated to design the hardware and Software platform. The embedded designed system is accommodate perform processing an image with digital FIR filter.

Hardware Platform

The hardware part of the designed embedded system is shown in Figure1. Base system builder tool is used to construct the system The hardware part of the system is designed has microprocessor hardware specification file MHS while the software part is described by the microprocessor software specification file MSS.

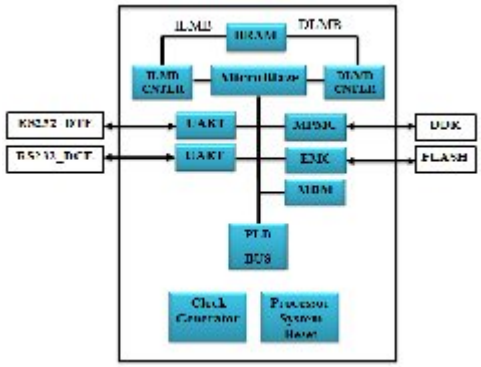

Figure 1 The block diagram of the system to be designed

The Memory can reside on-chip and off-chip. The on-chip memory as block RAM(BRAM) can be accessed by microblaze using Local Memory Bus(LMB). which is composed of two ports for data and instruction (DLMB), (ILMB) [4]. A general purpose interface known Processor Local Bus(PLB) provides bus infrastructure for connecting microblaze with both on-chip and off-chip memory and other peripheral connected to the system[5].

The Intel flash PROM with 16 Mbytes is used for storing image samples and the microblaze processor access it using External Memory Controller (EMC)[6] , while DDR-SDRAM with 64 Mbytes is used for program execution and is it is accessed by microblaze using Multi-Port Memory Controller (MPMC)[7].

The Universal Asynchronous Receiver Transmitter (UART) Lite Interface connects to the PLB and provides the controller interface for asynchronous serial data transfer[8]. The addresses map of the peripherals is shown in figure 2.

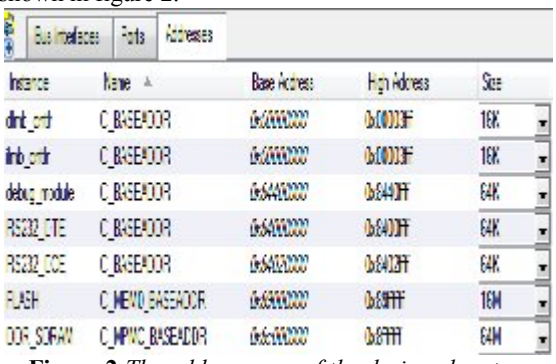

Figure 2 *The address map of the designed system components*

Software Platform

Now that the hardware design is completed, the next step is to define the software platform. Library generation (LibGen) tool creating the BSP which includes device drivers, libraries, configuring the STDIN/STDOUT, interrupt handling routines and other information related to the software part of the hardware platform elements generated with the XPS tool[9].

The software application is the code that runs on the hardware and software platforms. The source code for the application is written in a high level language such as C or C++, or in assembly language. The application is written in C language and oriented to image processing target.

Application Program

The application program represent processing certain image using image filter (High pass filter). Figure 3 display the application program.

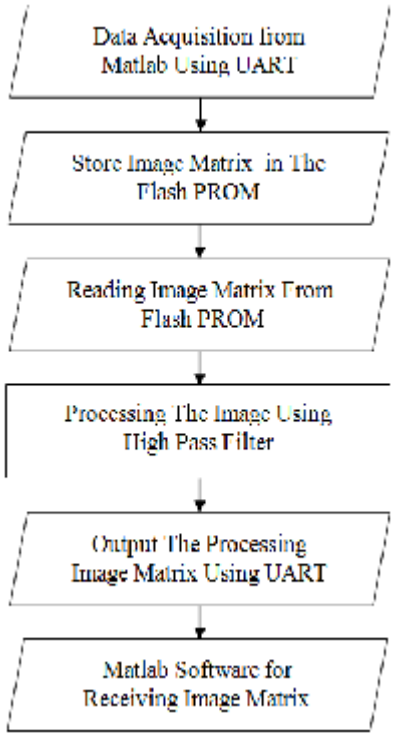

Figure 3 block diagram of the application program

A. Data Acquisition From Matlab Using UART

The image needed to be processed is acquired from Matlab, the image type is selected to be grayscale image, in the form of a data matrix whose values represent intensities with class type of uint8. The image data are sent from matlab to FPGA using fwrite() function provided by Matlab. The image data are received by processor system via UARTLite peripheral, using certain software driver as The XUartLite_Initialize(), XUartLite_ResetFifos() and XUartLite_RecvByte () functions[9]. The mentioned drivers are used in conjunction with C language in order to let the hardware interface of the UARTLite be seen by the C language.

B. Storing The Image In The Flash PROM

The received image matrix need to be stored in flash PROM, the space of PROM used to store image must be erased first. After erasing is completed the next step is writing the image to the flash PROM. The XIo_In8() and XIo_Out8() software driver are used for the purpose of Read/Write operations[9].

After erasing the needed section of flash memory image matrix element is written to it. Figure 4 shows the erase and write operation scheme[10].

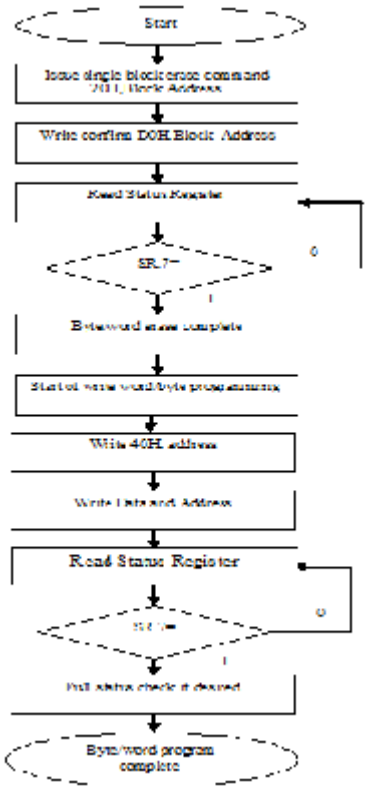

Figure 4 Flowchart of single-word/byte erase and programming

C. Reading Image From Flash PROM

The code image is read from the flash PROM using XIo_In8() and XIo_Out8() software driver[9]. The read array is confirmed by sending single command to the flash PROM. Write FFH to the device address place the flash PROM in read array mode ,subsequent reads output the data on data bus.

D. Processing The Image Using Digital Filter

The image matrix is now ready to be processed using high pass filter. linear filtering of an image is accomplished through an operation called convolution. Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is called the convolution kernel, also known as the filter. A convolution kernel is a correlation kernel that has been rotated 180 degrees.

One of many liner filtering is high pass filter. High pass filtering is accomplished using a kernel containing a mixture of positive and negative coefficients. An omnidirectional high pass filter– that is, one whose response is the same, whatever the direction in which grey level varies- should have positive coefficients near its centre and negative coefficients in the periphery of the kernel. The Kernel of the selected filter is[11]

$$H=\begin{bmatrix}-1 & -1 & -1\\ -1 & 8 & -1\\ 1 & 1 & 1\end{bmatrix}$$

The sum of the coefficients in this kernel is zero. The result of convolution is zero or some very small number. However, when grey level is varying rapidly within the neighborhood, the result of convolution can be a large number. This number can be positive or negative, because the kernel contains both positive and negative coefficients. High pass filters are useful in extracting edge and in sharpening[30].

The 'high boost filter' can be used to sharpen an Image. High boost filtering can be performed in a single convolution operation, using the kernel[11]:

$$H=\begin{bmatrix}-1 & -1 & -1\\ -1 & c & -1\\ -1 & -1 & -1\end{bmatrix} \quad (c>8)$$

When the central coefficient, c, is large, convolution will have little effect on an image. As the coefficient c gets closer to 8, however, the degree of sharpening increases.

The processed image data are sent to Matlab via UARTLite peripheral using the software driver XUartLite_SendByte(). The Matlab media acquire the data using fread() function.

Results and Discussion

The image matrix is acquired by the designed system from the Matlab program via UART DCE terminal. The image matrix is stored in the flash PROM and then processed by the system, the resulting of image matrix is transferred to the Matlab via DCE terminal.

The erase and write operations signals are debugged using chipscope analyzer, figure 5 shows the erase of first sector of flash PROM.

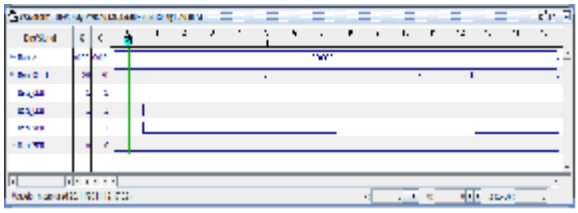

Figure 5 Erase flash PROM of first sector

Figure 6 shows the write operation of the image matrix sample on flash PROM, the write operation is done when MEM_CEN and MEM_WEN signals are active low and MEM_OEN active high . Figure 7 shows the read operation of the sample image monitored on the PLB bus.

Figure 6 Sample of flash write operation

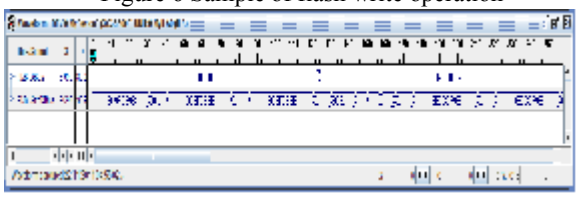

Figure 7 Sample of flash image matrix read operation

Masks1 is used for high pass filtering, Mask2 is unsharp mask for sharpening an image.

Mask 1

$$H=\begin{bmatrix}-1 & -1 & -1\\ -1 & 8 & -1\\ -1 & -1 & -1\end{bmatrix}$$

Mask 2

$$H- \begin{bmatrix} -1 & -1 & -1 \\ 1 & 9 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

Figure 8 provides examples of high pass filter on a cameraman image(256 X 256) with the Mask1. Figure 9 shows sharpen image processed with unsharp mask.

original image

Mask1

Figure 8 High pass filtering with 3 × 3 masks on the cameraman image

Orignal image

Unsharp mask

Figure 9 Unsharp filtering with 3 × 3 mask on the cameraman image

Conclusions

FPGAs can be adopted to configure an image processing system using embedded design technique. High storage element as DDR SDRAM and flash PROM are necessary to meet the high data density of the images resulting in a small area. Light weight portable image processing system.

References

[1] F. P, B. F. and Z. G. V. (2005)."Experiences with Soft-Core Processor Design". *IPDPS '05 Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, 4-8 April*, pp 167b.

[2] I. L., C. I., A. Gontean and M. C. (2010) " EDK Implemented Temperature Controller", IEEE, International Spring Seminar on Electronics Technology (ISSE), 12-16 May, pp 344 - 349

[3] Xilinx, Inc. (2007), "Processor Local Bus (PLB) v4.6 (v1.02a)", DS53.

[4] Xilinx, Inc. (2006). " LMB BRAM Interface Controller (v2.10a)", DS452.

[5] Xilinx, Inc. (2008)." Multi-Port Memory Controller (MPMC) (v4.00.a)". DS643

[6] Xilinx, Inc.(2010). "XPS Multi-CHannel External Memory Controller (v3.01a) ". DS575.

[7] Xilinx, Inc. (2008). "XPS UART Lite (v1.00a)", DS571.

[8] Xilinx, Inc.(2008) " EDK Concepts, Tools, and Techniques". XTP013 EDK 10.1.

[9] Xilinx, Inc. (2004). " Xilinx Device Drivers ". Documentation.

[10] Intel. (2005)." Intel Embedded Flash Memory (J3 v. D)". Datasheet.

[11] E. N. (2000)." Digital Image Processing". pp 158-159, First Edition, Addison-Wesley, United States of America.

تصميم نظام لمعالجة الصور على البوابات القابلة للبرمجة حقليا باستخدام تقنيات الانظمة المطمورة

مازن رجب خليل **سارة عبد الستار محمد**

E.mail: mazin_r_khalil@yahoo.com

الخلاصة

يهدف العمل إلى تصميم نظام لمعالجة الصور تم تطبيقه على شريحة البوابات القابلة للبرمجة حقليا (FPGA) باستخدام تقنيات الأنظمة المطمورة. تم إضافة ذاكرة غير المتطايرة نوع (Intel StrataFlash Parallel NOR PROM) للنظام لزيادة قابلية النظام لخزن عدد ضخم من عينات الصورة.تم استخدام مرشحات رقمية للتمرير العالي كوسيلة لمعالجة الصور كما تم استخدام لغة C لبرمجة النظام المطمور والتعامل مع الذاكرة المتطايرة.