International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

FPGA Based Real Time Object Tracking using Microblaze Core

Suraj Singh[Å*], Ajinkya Talele[Å] and Dipakkumar Wani[Å]

[Å]Electronics and Telecommunication, P.ES Modern College of Engineering, Shivajinagar, Pune, India

Accepted 07 May 2014, Available online 01 June2014, **Vol.4, No.3 (June 2014)**

Abstract

*This paper presents the image processing system based on Field Programmable Gate Array (FPGA) which is capable of tracking real time object using micro blaze as a processor. The presented work combines both hardware and software flow in which image acquisition algorithm is implemented as hardware module while other part like image processing is targeted to software. In our work we are using the standard video format of 640*480.we are processing the image on a soft core processor Micro blaze. The main idea of this work is to perform both image acquisition and image processing parallelly to increase the speed and efficiency of the system.*

Keywords: *FPGA, Xilinx, Micro blaze, object tracking.*

1. Introduction

Video processing has become an important application domain for smart embedded system. Video processing plays very crucial role in many applications such as human computer interaction, augmented reality, security system, video surveillance, automatic traffic control and many more, As for each of the area object tracking is fundamental part of the system. The objective of object tracking is to associate target object in consecutive video frames or for identifying the trajectory of moving object in continuous video frame sequence. To track the object successfully and efficiently real time processing is necessary in this application. Also real time processing increases the quality and reduces the amount of data to be transmitted or stored. Moreover, real time video processing is computationally demanding but often highly parallelizable, making it amenable to hardware implementation.

Proposed system is implemented on FPGA due to its ability to exploit the inherently parallel nature of many vision algorithms. FPGA is gaining more importance in embedded system due to its capability and availability of powerful Electronic Design Automation (EDA) tools. Now days FPGA consist of very high recourses so that it can handle and process many complex designs, so that no external recourses are required.

As the complete system design is being implemented on FPGA, the hardware software co-design concept is used; complex part or the system demanding processing power and parallelism is implemented on hardware. In proposed idea, image acquisition block consist of cmos camera and ram targeted on hardware while the processing core is targeted on software.

*Corresponding author: **Suraj Singh**

This paper consists of the following sections: 2.Brief description of complete system architecture, 3. System Analysis, 4. Result and 5.Conclusion.

2. System Architecture

We have proposed the embedded architecture for image and video acquisition and processing module for the proposed system. The design facilitates the streaming of video from camera to monitor through DDR memory and the FPGA logic in real-time. Proposed system architecture is shown in Fig1. It consist of Omni vision OV5642 5MP Camera, the Xilinx Spartan 6 FPGA board, video DAC ADV7218 and a VGA monitor to display the output.

The OV5642 is the 5 megapixel SOC camera sensor; it provides the full functionality of a complete camera on a single chip, including anti-shake technology, auto focus control, MIPI and high definition support (720p and 1080p).

The Spartan 6 (XC6SLX25) FPGA has mainly the following resources: 24051 logic cells, 3758 CLB (configurable logic blocks) slices, 38 DSP48A1 Slices and 936Kb of Block RAM (BRAMs), Input-Output Blocks (IOB), Digital Clock Management (DCM), Hardware Multipliers and Micro blaze soft processor core.

In this architecture Micro blaze embedded processor is used for the interfacing of FPGA-based custom modules and IPs along with the configuration of platform peripherals. The software environment of the system consists of application software and device drivers. The hardware part of the system includes the configurable logic blocks in FPGA. This integration of software and hardware provides the complete system functionality.

To display the output video DAC ADV7218 is used which is a triple high speed, digital-to-analog converter on a single monolithic chip. It consist of three high speed 10-

bit, video D/A converter with complementary output, a standard TTL input interface, and a high impedance, analog output current source. Video DAC reads the data from DDR on its 10-bit data line gives the analog video output which is connected with the TV tuner card to display the output on monitor.

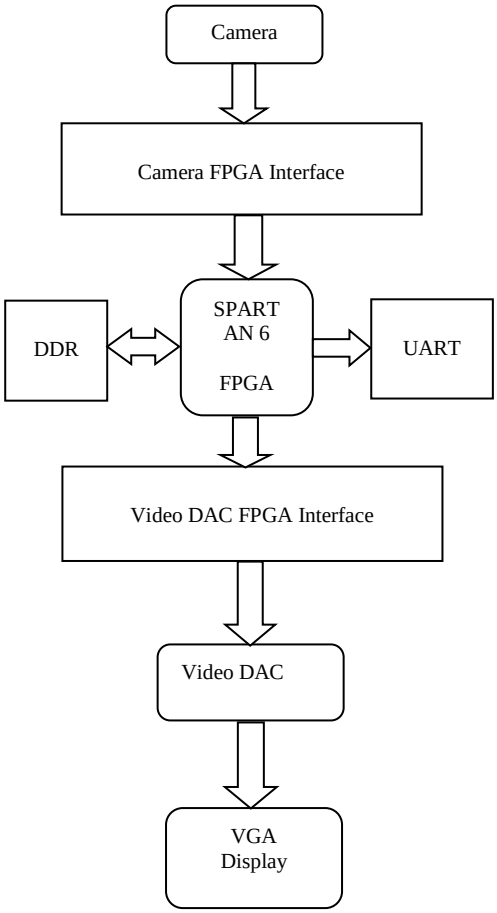

Fig.1 System Architecture

2.1 Image Acquisition

This part of the architecture is implemented as hardware module on Spartan 6 board. In proposed system input source is camera which gives the RAW image in RGB format of resolution 640x480. DDR is used as an external memory to buffer the video, for each input pixel corresponding ram address is generated and pixel information is stored in DDR. In proposed design only G-plane is stored in the memory to reduce the memory and computational overhead. The stored image in memory is forward to the processing block which is a soft core processor interfaced by General Purpose IO with the hardware module.

2.2 Image Processing

This part of design consists of embedded architecture of soft core processor Micro blaze which is implemented with the help of Xilinx Platform Studio (XPS) . Also in this architecture UART and GPIOs are implemented to interface the processor with the hardware module. Data from the SDRAM is read with the help of GPIO and then it is processed, Application program for the proposed system is written in Xilinx SDK.

3. System Analysis

As shown in Figure1, the input video generated by Image sensor has been transferred to Camera Input IP Core. This IP core decodes the format generated by camera, to generate synchronization signals like horizontal synchronization signal (HSYNC), vertical synchronization signal (VSYNC). Using HSYNC and VSYNC valid frame is written in the buffer memory, after every VSYNC pulse writing of one frame is completed. Every frame is then read by the Micro blaze from buffer and then it is processed. In our design video DAC is used to display the streaming video. The output module is implemented as hardware module. It gives the video output by reading the video from SDRAM using its 10-bit data line.

According to our application program whenever any object having gray level in desired range its location is determined and its centroid is calculated. The co-ordinate of the calculated centroid is send to the output module which marks a cross-hair on the centroid of the detected object. So, whenever the object moves or change its location the resultant cross-hair will also follow its centroid.

The output of video DAC is composite video signal (CVS) is interfaced with TV tuner card to which video cable is connected with CVS and is connected to the computer to show the video on monitor.

4. Results

The real-time video is captured from the camera. The captured video is color space converted by the hardware image acquisition block. The gray level video is converted into the frames and buffered into DDR SDARM memory.

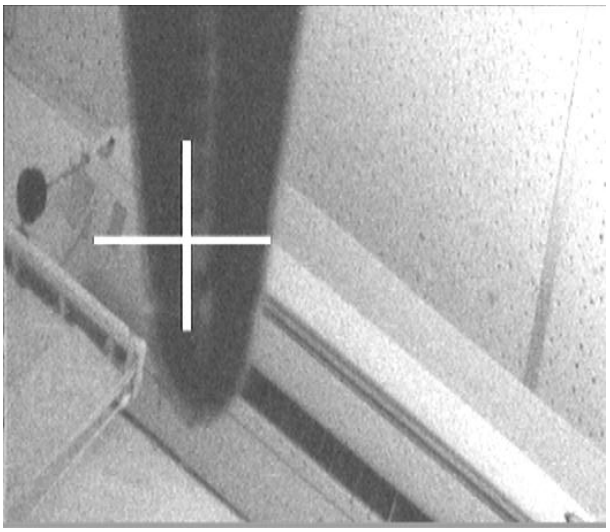

Fig.2 Object detected and marked with cross-hair.

The stored frames are processed and displayed on monitor having resolution of 640X480. The architecture uses

Xilinx Spartan-6 FPGA board. The embedded Micro blaze processor is used to process the image. A captured video frame with the detected object and centroid is marked with cross-hair is shown in "Fig. 2".

We have used the ISE, EDK and SDK Xilinx tools, version 13.1. Table 1 gives a summary of the resources consumed in the FPGA. The device utilization summary shows that, with the proposed embedded architecture based video acquisition and processing modules, the remaining FPGA resources are sufficient for implementing many practical real-time video processing applications.

Table 1 Summary of internal FPGA resources used

Number of Slices	1,062 out of 3,758	28%
Number of Slices Flips Flops	2,236 out of 30,064	7%
Number of 4 input LUTs	2,328 out of 15,032	15%
Number of bonded IOBs	106 out of 186	56%
Number of BRAMs	32 out of 52	61%
Number of DSP48A1	3 out of 38	7%

Conclusions

We have effectively implemented image processing algorithm in FPGA that shows considerable processing speed enhancement than traditional DSP. The idea of partitioning the object tracking algorithm into hardware-software co design has been implemented and found to be very effective in performance. Results obtained in this work compared with existing works, proves that the concept has reduced the processing delay of the real time video frame. The presented work may be extended by designing dedicated custom IP core for complete algorithm and software may be used only to control the IPs as well as display resolution. After implementation of complete algorithm in hardware, partial reconfiguration method may be used to minimize the utilization of device resource with negligible performance overhead.

Acknowledgment

We would like to thank our project guide Mrs. M.S.kanitkar (P.E.S.Modern College of Engg, Pune) for giving guidance and support to complete this work. At last but not least, we would like to thank Mr. Vaibhav Saini, Scientist-'D', ARDE-DRDO for his valuable suggestion and time to time discussions.

References

A.Oetken, S.Wildermann, J.Teich, D.Koch (2010) ,"A Bus-based SoC Architecture for Flexible ModulePlacement on Reconfigurable FPGAs," International Conference on Field Programmable Logic and Applications.

Alper Yilmaz, Omar Javed, Mubarak Shah (December 2006), "Object Tracking- A Survey", ACM Computing Surveys, vol. 38, No. 4, Article 13, pp.145-190

OV5642 Omni vision camera http://www.ovt.com/products

Spartan-6 Industrial Video Processing Kit – EDK Reference Design Tutorial, www.em.avnet.com/spartan6video.

Microblaze_Referance_Guidehttp://www.xilinx.com/support/documentation/sw_manuals/mb_ref_guide.pdf

http://www.xilinx.com/tools/sdk.htm