

Realtime Video Display

CSEE 4840 Project Proposal - March 2009

Manish Sinha, Srikanth Vemula, William Greene
Department of Electrical Engineering
Columbia University
{ms3766,sv2271,wmg2110}@columbia.edu

ABSTRACT

Displaying real time video captured from a camera is an essential function in a variety of applications ranging from CCTV security monitoring to webconference meetings. In this project, we propose to build a system on a FPGA, in conjunction with a commercial web camera, that can read a stream of video from a camera, decode this stream, and then display it on a local screen.

1 INTRODUCTION

The first phase of our project will center on interfacing with the camera. We intend to purchase a web camera with a USB output. We will use the USB port on the Altera DE2 board to read in the stream of data from the camera.

The second phase of our project will center on decoding the JPEG image stream. We will have to make careful design decisions including whether to handle JPEG decoding in hardware or in software. Satisfying the real time constraint will be a key challenge.

The third and final phase of our project will involve displaying the stream of decoded JPEG images on a screen. We intend to use the VGA interface on the Altera DE2 board and a connected LCD monitor to accomplish this.

2 RELATED WORK

Groups in past CSEE 4840 Embedded Systems courses have completed projects that overlap with our proposed one. In the 2007 class, the *Imagic* group [1] created a digital picture frame which needed to decode and encode JPEG images. In the 2006 course, the *spycam* group [2] displayed video captured from an internet camera with an

Ethernet interface. This group too had to handle JPEG decoding.

3 MAIN DELIVERABLE

We will deliver a system that, in real time, will: read the output data stream from a camera, decode this stream and, display it on a screen.

4 PLAN OF ACTION

1. Research web cameras and find which one is the easiest to communicate with.
2. Research the USB protocol [3] and start writing a driver to interface with the web camera.
3. Implement the IDCT [4] (for JPEG decoding) either manually or by using an existing library. Analyze performance between software and hardware implementations.
4. Write the code to display just one decoded JPEG image.
5. Write the code to display the series of decoded JPEG images and verify if the real time constraint is satisfied.

5 MILESTONES

- **March 30:** Complete the USB driver to interface with the web camera and read the raw stream of data.
- **April 13:** Complete decoding and display of a single JPEG frame.
- **April 29:** Complete decoding and display of a series of JPEG frames.

6 IF TIME PERMITS

If we finish early, we will extend our system to include streaming functionality over a local area network so that a destination system could view the video being captured from a source system.

7 REFERENCES

1. CSEE 4840 Spring 2007 - Imagic:
<http://www1.cs.columbia.edu/~sedwards/classes/2007/4840/reports/Imagic.pdf>
2. CSEE 4840 Spring 2006 - spycam:
<http://www1.cs.columbia.edu/~sedwards/classes/2006/4840/reports/spycam.pdf>
3. USB 2.0 Specifications:
http://www.usb.org/developers/docs/usb_20_122208.zip
4. IDCT:
<http://vsr.informatik.tu-chemnitz.de/~jan/MPEG/HTML/IDCT.html>