BufferCam

Video Recording "Black Box" for Bicyclists

ECE4007 Final Report

Section L01 Fall 2008

Derek Greene

David Ryan

Andrew Yang

Project Advisor:
Aaron Lanterman

Georgia Institute of Technology

Table of Contents

Executive Summary

The Buffercam is a video recording "black box" designed for bicycles, cars, and motor vehicles. The camera continuously records video to a microSD flash memory card. If the vehicle is involved in an accident, the device will automatically preserve video footage of the thirty seconds before and after the accident's onset. The automatic recording will be triggered by an accelerometer, similar to the ones used for triggering airbag deployment in automobiles. In addition, the device will feature a pushbutton so that events can be recorded manually at the touch of a button. For example, video can be captured if the user witnesses an accident, a crime, or suspicious behavior.

The Buffercam provides video evidence for determining the at-fault party in motor vehicle accidents. The video can also be used as evidence in injury compensation lawsuits, insurance claims, and charges of criminal negligence. The Buffercam also helps deter "hit-and-run" accidents as the drivers become aware that they could be caught on camera and face increased charges. This device can also save the user medical costs by shifting the financial burden to an at-fault driver.

A production version of this device would cost under $50, making it affordable for all types of vehicles. In addition, multiple Buffercams may be used on a single vehicle to provide even better video coverage.

1. Introduction

The Buffercam is a video recording device used to capture video of events to be used later as potential evidence of wrongdoing in a traffic accident. It has other applications for casual use in capturing day to day occurrences. The camera constantly records video similar to store surveillance cameras; however, the Buffercam is capable of tagging events within the video footage for easy retrieval. These video events can be captured manually or automatically.

1.1 Objective

The Buffercam provides video evidence for determining the at-fault party in motor vehicle accidents. The video can also be used as evidence in injury compensation lawsuits, insurance claims, and charges of criminal negligence. The Buffercam also helps deter "hit-and-run" accidents as the drivers become aware that they could be caught on camera and face increased charges. This device can also save the user medical costs by shifting the financial burden to an at-fault driver.

1.2 Motivation

Each year hundreds of accidents involving motor vehicles and bicycles are not reported. Even worse, many at-fault motorists often are not held accountable for their actions. In New York, the most motorists at fault in killing pedestrians and cyclists did not even get a ticket [1]. Police rely on evidence on the scene, witnesses, and the parties involved determining whether to issue a citation; because all these are circumstantial, officers sometimes do not file reports [2]. The Buffercam would provide objective evidence to law enforcement to allow better decisions to be made.

1.3 Background

The Buffercam is similar to the cockpit voice recorder (CVR) and the flight data recorder (FDR) on airplanes. The National Transportation Safety Board requires at least two hours of recording capability on its CVRs [3]. Level-3 Aviation Recorders makes flight data recorders capable of recording over 25 hours of flight data [4].

The most important pieces of the Buffercam are the camera sensor and the storage medium. There are several types of camera technology; the two most popular types are charge-coupled devices (CCDs) and complementary metal-oxide-semiconductor (CMOS). CMOS has the advantage of low power consumption and high noise immunity [5]. The storage medium for data loggers are typically flash memory. Flash memory is different from random access memory (RAM) in that flash memory is not volatile, so retains its data after power is turned off. Both RAM and flash memory are considered solid-state storage, since they require no moving parts. In contrast, hard drives are mechanical devices that are prone to being damaged by vibration and shock. Conventional hard drives also require power for an electro-mechanical motor, while flash memory only requires power to perform a read or a write. RAM only requires a small amount of power, at a short interval to maintain the data in the devices. As a result, RAM and flash memory require less power than a conventional hard disk so they can provide longer battery life.

2. Project Description and Goals

The production cost was limited to $50, making it affordable for all types of vehicles. The Buffercam records thirty seconds of video before and after a triggering event. The Buffercam automatically triggers itself via an accelerometer and is able to be manually triggered by a

pushbutton. The video is easily retrieved off the memory card. In addition, the device is small and lightweight and consumes a minimal amount of power to maximize battery life.

3. Technical Specifications

The target video resolution for the Buffercam was 320 by 240 pixels. This resolution was exceeded in the prototype, which had a resolution of 352 by 288 pixels. This resolution exceeds the standard video resolution for security cameras [6]. Although the target recording rate was at least 7.5 Frames per Second (FPS), the prototype was only capable of a frame rate of 2.8 frames per second. The primary reason for this shortfall was the write speed to the microSD card using SPI mode. After an event is triggered the video is buffered onto a microSD card. The system detects when an impulse is applied to it and automatically triggers the recording process. The recording process is also able to be triggered by a pushbutton. Final post-processing is done on the computer and saves the video using the WMV7 codec.

The prototype produced incorporated many aspects of the final design. However, a production design would include a fabricated PCB and a durable enclosure. Buffercams that are used in motor vehicles may be powered using the vehicle's battery. For other vehicles, the Buffercam may be powered by its own battery. In this case, the Buffercam should be able to charge off of a 5VDC wall outlet or a USB port and provide at least three hours of battery life. A summary of specifications is given in Table 1.

Table 1. Specifications

Recording	
Resolution	352x288 pixels (CIF)
Frame Rate	2.8 FPS
File Format	
On Card	BMP
Final	WMV7
Form Factor	
Diameter	1”-2”
Length	6”
Power	
Battery Life	>3 Hours
Voltage	3.7V
Battery Type	Li-Ion Polymer
Recharge	USB or 5VDC

4. Design Approach and Details

4.1 Design Approach

The Buffercam is composed of a PIC microcontroller surrounded by several peripheral components and modules (see Figure 1). These auxiliary devices include a camera, accelerometer, FPGA, First-in-First-out (FIFO) memory chip, and a microSD card slot. These auxiliary devices can be grouped into three main categories, image acquisition, storage, and triggering.

The PIC32 was selected as the core of the Buffercam since it has many of the interfaces that are used to communicate with the peripheral devices, including I^2C, SPI, and PMP. The PIC32 is the most powerful PIC processor from Microchip and runs at 80MHz; however, it consumes very little power. It is easily programmed in C through Microchip’s MPLAB Integrated Development Environment and Microchip’s C32 compiler.

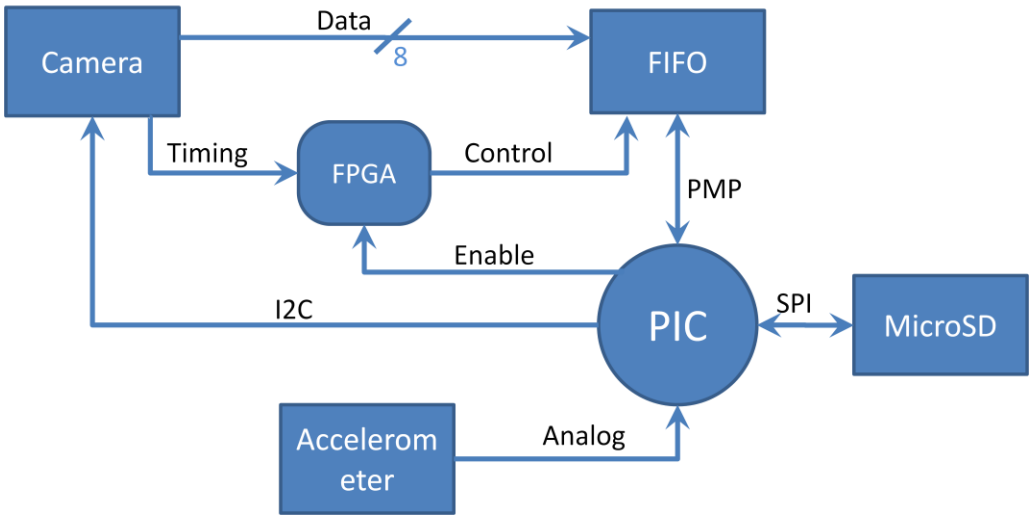

Figure 1. Basic Buffercam structure.

4.1.1 Image Acquisition

Image acquisition involves the camera, FIFO, and FPGA. The camera used in the Buffercam prototype is a Toshiba TCM8240MD camera module that supports up to a 1.3MP resolution and 15 FPS (shown in Figure 2). Because the camera was a unique quad package a custom breakout board was required to access the pins on the camera. The camera can output data in RGB, YCbCr, and JPEG formats.

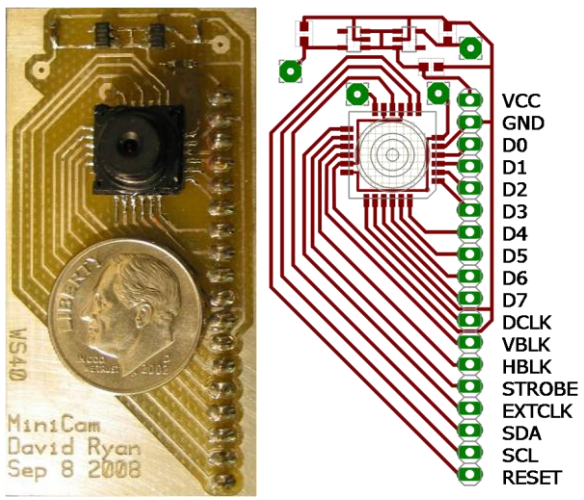

Figure 2. Camera with custom breakout board.

The camera's setup is controlled through the I^2C bus from the PIC32. Unfortunately the data coming out of the camera cannot be paused and the time allowed to get a bit of data before it disappears is very short. The solution to this problem is to have a buffer between the camera and the PIC32. The Averlogic AL440B-12 4MBit FIFO chip was chosen to be our FIFO memory. It allows asynchronous reading and writing to the chip which allows the camera to write to the memory at different rate than that which the PIC32 is reading off it. Because the FIFO chip had a pin spacing of 0.8mm an adapter board could not be found. A solution to this was found by taking a 0.75mm adapter board and tilting it. This allowed the pins to line up on the board. A picture of this is shown in Figure 3.

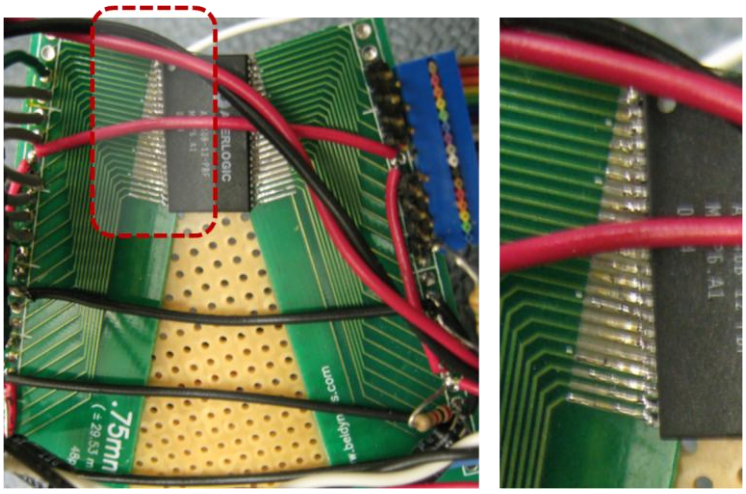

Figure 3. FIFO with adapter (left) with pins zoomed in (right).

The FPGA is essentially being used as a triple-input AND gate which handles the timing of the writing from the camera to the FIFO; it governs when data should be allowed to enter the FIFO. The relevant timing signals coming out of the camera are the horizontal blank (HBLK) and vertical blank (VBLK) signals. They signal when the camera is in the HBLK and VBLK.

The vertical blank frames an image and within the vertical blank the image data is only transmitted during the horizontal blank. An example of this is shown in the timing diagram in Figure 4. The signals VBLK, HBLK, and a third, FIFO_ENABLE, from the PIC32 go into the FPGA. The output of the triple-input AND gate made by the FPGA goes to the FIFO chip's write enable pin. When all of these signals are high, data is allowed to go into the FIFO.

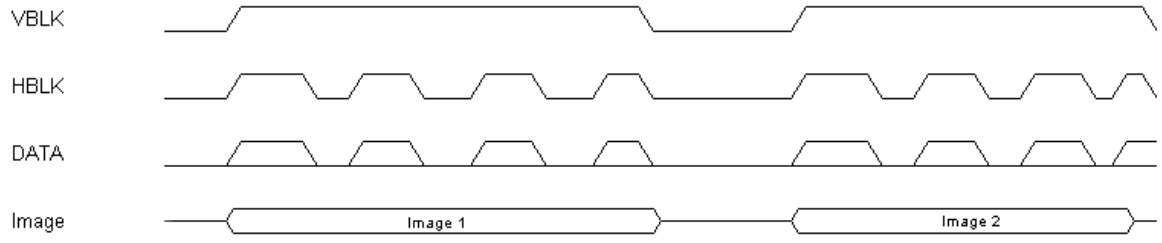

Figure 4. Timing diagram of VBLK and HBLK corresponding with image data.

Future work includes replacing the FPGA with a single triple-input AND gate chip. This was attempted, but was not successful because the propagation delay was too great. The propagation delay of the gate tested was near 100ns; the delay needs to be less than 50ns to operate properly with a 10MHz data clock. For JPEG data needs the delay needs to be further reduced to 25ns to accommodate the 20MHz output clock. The Fairchild 74AC11 meets the specifications for propagation delay and should be considered for future work. When HBLK and VBLK go high the data needs to be clocked into the FIFO before the falling edge of the clock, otherwise the byte of data may not be read correctly. This is shown in timing diagram in Figure 5.

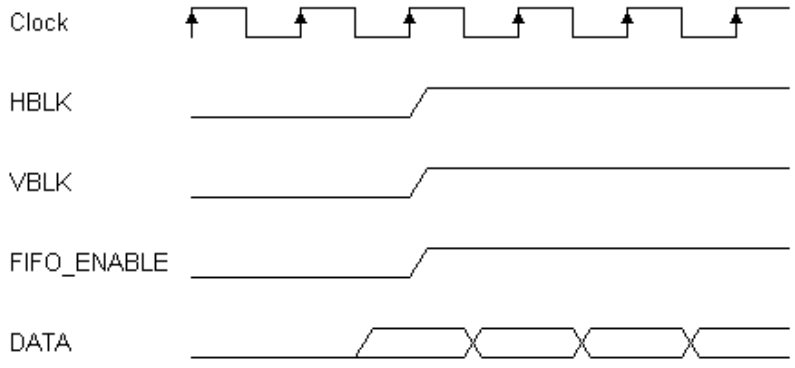

Figure 5. HBLK, VBLK, and FIFO_ENABLE signals into the triple-input AND gate.

The data is read off the FIFO through the PIC32's Parallel Master Port (PMP) bus. The PMP bus is a highly configurable, eight or sixteen bit parallel data bus specifically designed to communicate with parallel devices. The PIC32 in the Buffercam prototype uses the PMP in eight-bit mode with no address lines and a read strobe attached to the read clock on the FIFO chip. When the PIC32 "strobes" the read line on the FIFO memory, the memory outputs the next byte of data. This data is latched onto the parallel data bus of the PIC32 and transferred into the memory of the PIC32.

Figure 7 shows a schematic of the camera, FIFO, and triple-input AND gate. All the signals that end in the "<<" symbol connect to the PIC32. The SCL1 and SDA1 signals are the I^2C signals for configuring and controlling the camera connected to the I^2C module on the PIC32. The OC4 signal uses the output compare of the PIC32 to generate a clock using Pulse Width Modulation (PWM) for the camera clock. The CAMERA_RESET signal resets the camera. The FIFO_ENABLE signal allows the PIC to control whether or not image data should be captured by the FIFO. The FIFO_IO_RESET signal resets the internal memory pointers of the FIFO to zero. The FIFO_GLOBAL_RESET signal is the power-on reset for the FIFO. PIC_PMPD is the

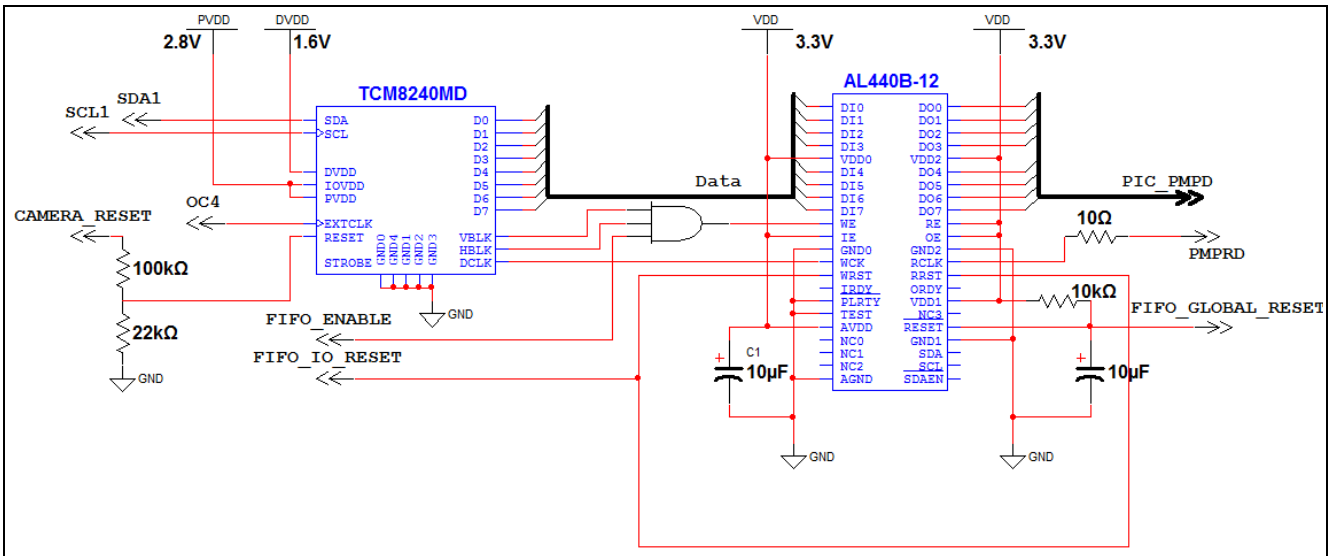

Figure 7. Image acquisition schematic.

data bus from the FIFO to the PIC32's PMP module; the PMPRD signal is the read strobe of the PIC32's PMP module.

There is an outstanding issue with a byte being dropped between the camera and the FIFO. This occurrence is random and occurs approximately once every 300 images. The effect of a missing a single byte is shown in Figure 6. When a byte is dropped, subsequent images will stay in the color-shifted state until another byte is dropped again. The image can be restored by removing or adding a byte after the bitmap header. A simple computer program was written to correct the color-shifted images.

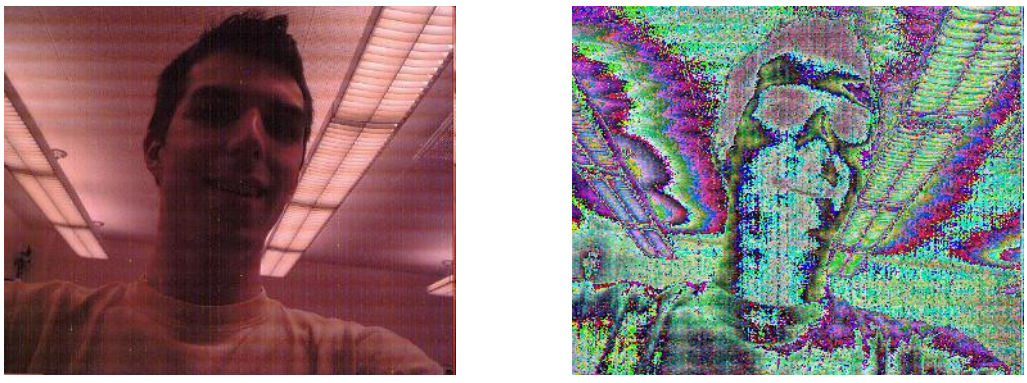

Figure 6. Original picture (left) with one byte removed (right).

4.1.2 Storage

The storage medium selected for the Buffercam is a microSD card. This format was chosen because it provides high capacity flash memory in a small form factor. It also uses a simple communications protocol that was easily implemented on the PIC32, Serial Peripheral Interface (SPI). MicroSD cards using SPI have a maximum speed of 25 MHz; however, due to limitations in the way the PIC32's 80 MHz clock can be divided, the prototype uses a 20 MHz SPI clock.

The file system used on the microSD card was FAT16. Two libraries provided support for this, Microchip's MDDS File System Library and the FatFs library. The Microchip MDDS library was much slower than the FatFs library. Microchip's library was only able to attain approximately 230 KB/s write speed to the card whereas FatFs was able to attain approximately 800 KB/s. This was tested by calling the fwrite() equivalent functions in both libraries on 4KB arrays writing that array 256 times for a final file size of 1MB. It was determined that this disparity in performance was because the Microchip library did not support optimizations when writing multiple blocks on microSD card, while the FatFs library supported writing multiple 512 byte blocks at once.

4.1.3 Triggering

There are two methods to trigger the Buffercam. The first is a simple pushbutton. The second is the accelerometer. The accelerometer is an Analog Devices ADXL321 dual-axis ±18g accelerometer that provides two analog outputs, one for each axis. The PIC32 reads the two analog inputs using its built in analog-to-digital converter. The numbers are then processed through an exponential moving average filter to smooth out the data and eliminate noise. The formula for the exponential moving average filter is shown in Figure 8. The effects of the smoothing are shown in Figure 9.

$$y[n]=\alpha \cdot x[n] + (1-\alpha) \cdot y[n-1]$$

$$\alpha = 2/(N+1)$$

Figure 8. Exponential moving average filter formula.

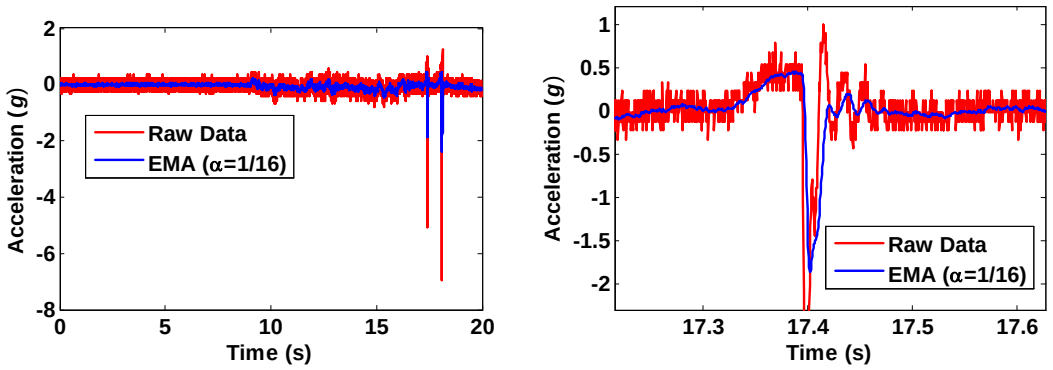

Figure 9. Effect of exponential moving average filter (left) with zoomed in area (right).

4.1.4 Post Processing

The post-processing was done with and AutoIt3, a windows script editor that allows a script to be compiled to an executable file. The script executes a program, Imagemagick, called mogrify which converts the bitmap images to PNG format. After this they are fed into mencoder which encodes the images into a video file.

4.2 Codes and Standards

There are several communications standards in use. Two very simple serial communications protocols are used, SPI and I^2C. SPI is used to transfer data to the microSD card and I2C is used to set up the camera. The PIC32 supports both of these standards with relative ease and includes a peripheral library to assist in the setup of these protocols.

The file system used on the microSD card is FAT16. This was chosen for its widespread adoption with support in virtually all computers. Many memory cards, memory sticks, etc. also use this file system.

4.3 Constraints, Alternatives, and Tradeoffs

The primary constraints are cost and power. Because this system is designed to be portable, battery life must be taken into consideration. For maximum market saturation the device must be affordable for drivers of all types of vehicles.

4.3.1 Linux-Based Microcomputer

A simpler alternative involves using a Linux-based microcomputer such as the "Gumstix" computer. This method allows all of the interfacing to be handled in high-level programming language instead of assembly or VHDL. Unfortunately, microcomputers like the Gumstix require a large amount of power; this power consumption is probably beyond the limit necessary to provide the desired battery life and small size. Also, the cost of such devices far exceeds the $50 target cost per device.

4.3.2 USB Flash Drive

A USB flash drive was considered as a storage medium. This was originally selected because the PIC32 supported USB host signals and because it is a small and standard medium. However, technical difficulties were encountered trying to get the USB flash drives to recognize as USB mass storage devices. This was attributed to the fact that most USB flash drives on the market are actually USB composite devices and not monolithic USB mass storage devices. The Microchip USB stack is primitive compared to the USB stacks on computers and was unable to recognize any USB flash drives attempted.

4.3.3 Structured ASIC

Alternative designs that were considered include an FPGA-only design using a system on a chip. While this would be the smallest design, it would require developing more code to handle the interfaces between the components. It may also draw more power and may required being designed for a structured ASIC device in order to meet the power requirements. This design is depicted in Figure 10.

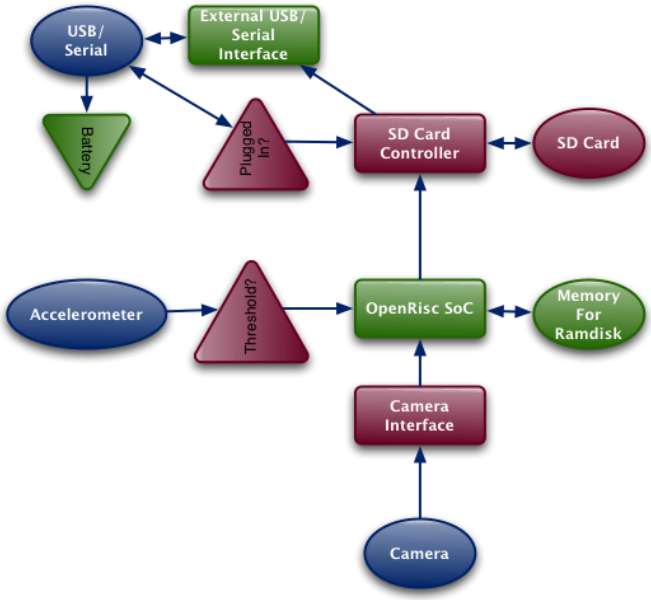

Figure 10. FPGA system on a chip design.

This approach uses a software processor on the FPGA to deal with data, and allowing software upgrades to improve functionally. Unfortunately, this design would require a FPGA with many more gates, which is more costly than using a smaller FPGA in combination with a PIC microcontroller. Furthermore, a lot of work would be required to adapt the available system on a chip core for the Xilinx FPGA. Additional work would be required to interface the camera and accelerometer to this type of system.

4.3.4 CMUcam3

The main setbacks encountered involved the camera. The datasheet for the Toshiba TCM8240MD camera module is hard to understand and incorrect in many instances. There are undocumented registers and some formulas in the datasheet are wrong. This led us to pursue an alternative camera, the CMUcam3 developed by Carnegie-Mellon University. It outputs frame data on a Serial UART bus which is a standard supported by the PIC32. However, the resolution supported by the CMUcam3 is not as high as the Toshiba Camera and transfers over Serial UART are much slower forcing us to use a slower frame rate.

This option was forgone, because through cooperation with several people online, some undocumented registers were found and an image was successfully retrieved from the Toshiba camera. During this process it was also determined that some of Toshiba's formulas for determining, specifically SPCOUNT. Although Toshiba's datasheet did not state what the SPCOUNT does, it was necessary, using trial and error, to get this value correct for proper camera operation.

5. Schedule, Tasks, and Milestones

The schedule originally proposed had many setbacks. The original schedule is shown in Figure 11. The camera was the primary source of the setbacks. The datasheet for the camera was extremely poorly written and we did not get any major breakthroughs getting the camera to work until mid-November when another developer managed to reverse-engineer a set of camera settings that worked. Interfacing the storage with the PIC took longer because a USB memory stick was also examined as a solution. The USB memory stick was ruled out as a storage medium at the beginning of November due to technical difficulties.

ID	Task Name	Duration	Predecesso
1	**Proof of Concept**	**55 days**	
2	Develop Buffering Algorithm	2 wks	
3	Assemble Camera Breakout Board	3 days	
4	Interface RAM with FPGA	2 wks	
5	Interface Camera with FPGA	3 wks	3
6	Interface SD card with PIC	2 wks	
7	Interface PIC with FPGA	1.2 wks	
8	Capture Image to SD Card	0 days	4,5,6,7
9	Test and Refine Buffering	5 days	8
10	Interface PIC with USB	1.5 wks	
11	Interface Accelerometer with PIC	1 wk	
12	Calibrate Accelerometer Threshold	2 days	11
13	Develop Real Time Clock on PIC	3 days	
14	Testing and Debugging	5 wks	12,13,9
15	Project Demo	0 days	14
16	**Useable Prototype**	**36 days**	
17	Develop battery charging circuit	3 days	
18	FPGA Configuration on Boot	1 wk	
19	Design PCB	1.5 wks	17,18,9,11
20	PCB Manufacturing	2 wks	19
21	Assemble PCB	3 days	20
22	Test and Debug PCB	0.5 wks	21
23	Assemble and Test Finished Product	5 days	22

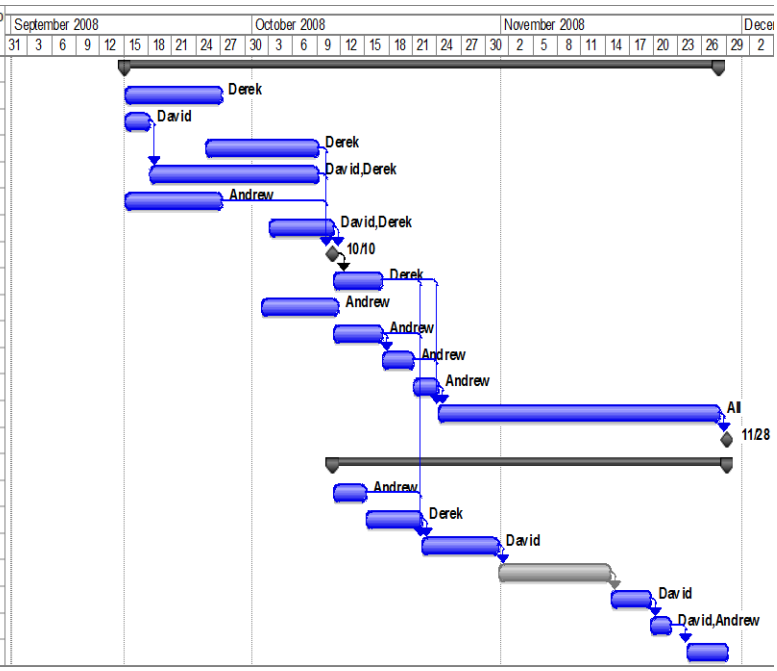

Figure 11. Gantt chart showing projected schedule.

6. Intercontinental Collaboration

This project was unique in that one team member was based in Metz, France for all but the final week of the project. This presented some challenges for communications, development, presentations, and testing.

6.1 Communications

Scheduling meetings was especially challenging not only trying to coordinate three busy schedules but also due to the six hour time difference between Atlanta and Metz. The meetings were unorthodox because of the use of online meeting tools such as Skype and Oovoo [*7*]. Furthermore, collaboration was facilitated using Google Groups, an online collaboration site helping manage emails, files, and information.

6.2 Development

Development was well coordinated by ensuring a well understood division of labor. However, it was made more difficult for the member in France because ideas could not be informally discussed outside meetings. The division of labor was also dictated by the limited resources available to the member in France as equipment, tools, and parts were not consistently duplicated. At the beginning of the semester, parts were sent via two Georgia Tech staff members traveling from Atlanta and Metz. However, as design changes were made, parts could not be acquired in France, therefore, considerable difficulties were experienced in the development process.

6.3 Presentations

Presentations were effectively executed using the Polycom video conferencing system normally used for distance learning. The member in France appeared on a high definition television in Atlanta next to the Atlanta group members and presented his part. Efforts were made to try to establish remote control of the computer handling the presentation using RealVNC [8], so that the group member in France may change the slides on his own and perhaps even change them for the Atlanta group members. However, this was unsuccessful due to the unreliable wireless connection on both ends.

6.4 Testing

Testing was also made difficult by not having duplicate equipment. At one point, the member in France had developed an FPGA design for a development board in Atlanta. This required the team member in the Atlanta to program the FPGA development board and, over a Skype video conference [9], report back data for the member in France to correct problems and bugs.

7. Project Demonstration

The proof-of-concept Buffercam has been demonstrated on prototype and demonstration boards connected together. A force was applied to the accelerometer to trigger the recording process. After the recording was complete the card was read using a computer and a computer program was run to convert the images to video. A manual pushbutton was also pressed to manually trigger the Buffercam. The demonstration showed the Buffercam's ability to capture the event thirty seconds before and after the event. A picture of the final demonstration's proof-of-concept device is shown in Figure 12.

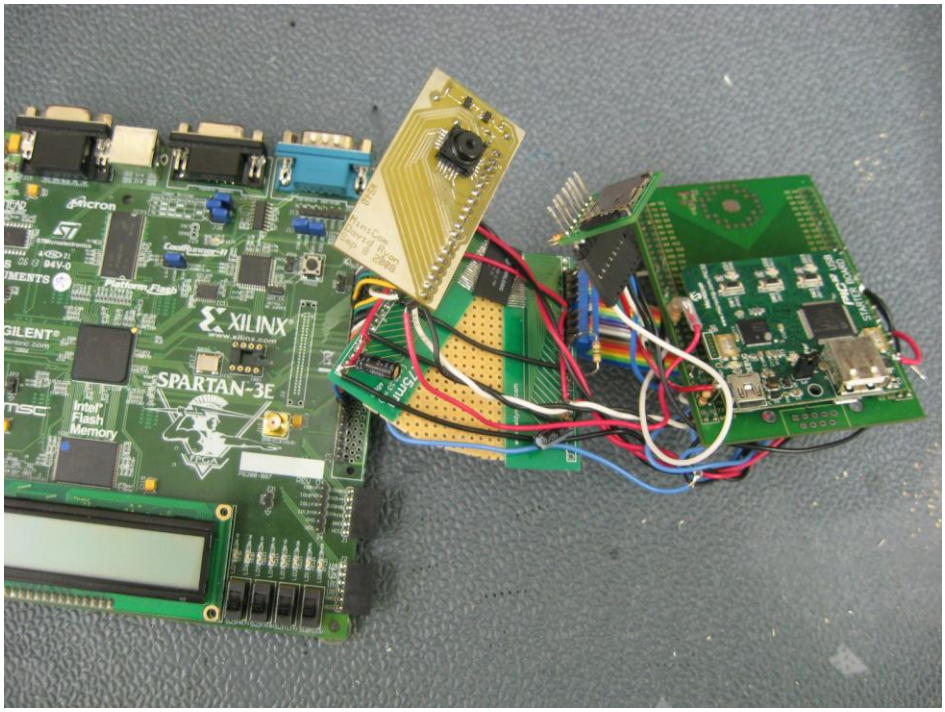

Figure 12. Final Project Assembly.

8. Marketing and Cost Analysis

8.1 Marketing Analysis

An estimated 91 million bicycling trips were made during the summer of 2002 in the United States. Of these trips, 48% were made on paved roads and an additional 18% were made on the shoulders or bicycle lanes of paved roads [10]. This represents a large target audience for the Buffercam device. Since many people are concerned with safety, statistics about cyclist injuries and fatalities can be used to market the product. For example, over 51,000 cyclists have died in traffic crashes in the United States since 1932; in 2006 alone, over 44,000 cyclists were injured in traffic crashes [4]. The device could be sold at bicycle shops and retail stores.

With a few slight modifications, the device could also be targeted toward several other markets. Since motorcycles suffer from many of the same safety problems as bicycles, the Buffercam could be sold to motorcyclists if a mount was designed to attach the camera to the vehicle. Moreover, the Buffercam could be used in cars and other motor vehicles with a dashboard mount. For this application, acceleration may not be adequate to detect minor collisions, but it could still provide the manual recording functionality. A small microphone could be added to the device to allow recording to be triggered by loud noises like car alarms.

8.2 Cost Analysis

Cost estimates were obtained for each of the buffercam parts from the suppliers listed in Table 2**.**

Table 2. Bill of Materials

Component Name	Details	Price Estimates
PIC MCU	PIC32MX440F128H-80I/PT	mouser.com/digikey.com
512K x 8bit FIFO	Averlogic AL440B-12	semiconductorsuperstore.com
1.3 MP Camera	Toshiba TCM8240MD	sparkfun.com
Accelerometer	Freescale MMA7360LT	digikey.com
USB Li-Ion Charger	Maxim MAX1555	maxim-ic.com
MicroSD Card Connector	3M 2908-05WB-MG	digikey.com
USB Connector	Hirose UX60-MB-5ST	digikey.com
PCB	2-layer, 9 sq. inches	4pcb.com/sunstone.com
Li-Ion Battery	RCR123A 800mAh (sku.722)	dealextreme.com
Mount	Universal Bike Mount (sku.8274)	dealextreme.com
Housing	Aluminum Flashlight (sku.484)	dealextreme.com
Assembly + Testing	20 Parts, 30 SMT, 3 Fine Pitch	screamingcircuits.com

The cost for materials, manufacturing, assembly, and testing is shown in Table 3.

Table 3. Estimated Material and Manufacturing Costs

	Quantity				
Component Name	**10**	**100**	**1000**	**10000**	**100000**
PIC Microcontroller	6.58	5.57	*5.00*	*5.00*	*5.00*
512K x 8bit FIFO	13.85	13.85	13.85	*12.00*	*10.00*
1.3 MP Camera	9.95	8.96	7.96	*7.00*	*6.00*
Accelerometer	4.04	3.06	2.82	*2.50*	*2.00*
USB Li-Ion Charger	1.34	1.34	0.85	0.85	0.85
MicroSD Card Connector	0.98	0.88	0.78	0.74	0.74
USB Connector	1.18	0.70	0.50	*0.45*	*0.45*
PCB	6.60	6.20	2.91	*2.58*	*2.25*
Li-Ion Battery	1.39	1.39	1.39	*1.20*	*1.00*
Bicycle Mount	1.47	1.37	*1.25*	*1.10*	*1.00*
Housing	6.73	6.73	4.00	*3.50*	*3.00*
Other Parts	*6.00*	*5.50*	*5.00*	*4.50*	*4.50*
Assembly + Testing	72.30	17.76	*10.00*	*8.00*	*6.00*
Material Cost Per Unit:	**$132.41**	**$73.31**	**$56.31**	**$49.42**	**$42.79**

9. Summary and Conclusions

The group has successfully demonstrated a working prototype of the Buffercam. The camera can successfully be triggered with the pushbutton or the accelerometer and records thirty seconds of video prior to and after the triggering event. The group has also successfully demonstrated the conversion to a video file. This prototype is not without bugs. The accelerometer's threshold needs to be adjusted and perhaps a static calibration can be set and from time to time adjusted by the user. In addition to this, there is a problem with occasionally dropping a byte of image data that causes the images to become color-shifted. Overall the demonstration met the criteria for success and revealed the feasibility and design issues associated with creating such a device.

References

[1] Bicycle Almanac. (2008) Bicycle Universe. [Online]. http://bicycleuniverse.info/transpo/almanac-safety.html

[2] D. Noyes. (2007, May) ABC7 News. [Online]. http://abclocal.go.com/kgo/story?section=news/iteam&id=5329449

[3] NTSB. (2007, Nov.) National Transportation and Safety Board. [Online]. http://www.ntsb.gov/recs/mostwanted/aviation_recorders.htm

[4] Level-3. (2008) Level-3 Aviation Recorders. [Online]. http://www.l-3ar.com/PDF_Files/MKT051_FA2100FDR.pdf

[5] DALSA. (2008) CCD vs. CMOS. [Online]. http://www.dalsa.com/markets/ccd_vs_cmos.asp

[6] Rockpile Security. (2005) CCTV Resolution and Detail. [Online]. http://www.rockpilesecurity.com/pdf/wp_cctv_resolution_and_detail_01.00.pdf

[7] Oovoo. Oovoo. [Online]. http://www.oovoo.com

[8] RealVNC. RealVNC. [Online]. http://www.realvnc.com

[9] Skype. Skype. [Online]. http://www.skype.com

[10] The Gallup Organization. (2002) National Survey of Pedestrian and Bicyclist Attitudes and Behaviors. [Online]. http://drusilla.hsrc.unc.edu/cms/downloads/NationalSurvey_PedBikeAttitudes_Highlights2002.pdf

Appendix: Code

The code is provided in the supplementary file "Buffercam_code.zip". Inside you will find:

Folder	Buffercam FPGA Code
Contains	The FPGA code from our final demonstration. Essentially the triple-input AND gate. The development board we used was the Xilinx Spartan 3E Starter Board.

Folder	Buffercam Oscilloscope Capture FPGA code
Contains	The FPGA code used to get the first image off the camera. At that time the PIC32's PMP bus was not interfacing with the FIFO yet. This FPGA code includes a state machine to manage to FIFO and to dump the data with the push of a button to the oscilloscope.

Folder	Buffercam PIC32 Code
Contains	The PIC32 code from our final demonstration. This does all the functions of the Buffercam. Includes writing to the SD card, capturing from the camera, checking the accelerometer, etc.

Folder	util
Contains	Image and video tools used by the video compiler to assemble the folders of images into a video file.

File	bin2data.m
Contains	MATLAB function for interpreting the binary data format of the MSO6000 series scope captured using the Intuilink Data Capture program to extract image.

File	FixBMPoffset.au3
Contains	Script that fixes color-shifted bitmaps.

File	Port Assignments.txt
Contains	Port assignments for the PIC32

File	Video compiler.au3
Contains	Source code for the video compiler

File	Videocompiler.exe
Contains	The video compiler executable. Use with "util" directory's contents.