12-2007

FPGA-based Image Analysis System for Cotton Classing

Muhammad Imran Sharafat
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Muhammad Imran Sharafat entitled "FPGA-based Image Analysis System for Cotton Classing." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Donald W. Bouldin, Major Professor

We have read this thesis and recommend its acceptance:

Gregory Peterson, Jack S. Lawler

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting here with a thesis written by Muhammad Imran Sharafat entitled "FPGA-based Image Analysis System for Cotton Classing". I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment for the degree of Master of Science, with a major in Electrical Engineering.

Donald W. Bouldin
Major Professor

We have read this thesis and
recommend its acceptance:

Gregory Peterson

Jack S. Lawler

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the
Graduate School

(Original signatures are on file with official student records.)

FPGA-based Image Analysis System for Cotton Classing

A Thesis Presented for
the Master of Science Degree
The University of Tennessee, Knoxville

Muhammad Imran Sharafat

December 2007

Acknowledgements

I praise and thank GOD for all the blessings and I ask for forgiveness for all my shortcomings. I am thankful to my mother and father for all their love and patience in dealing with me. I would like to thank my wife, Tessa (Aneesa) Sharafat, for her loving care and for supporting and encouraging me to get my Master's degree in Electrical Engineering. I also thank my daughter, Aminah Khan, and my boys, Abdul-Raheem Khan and Abdul-Mumin Khan, for their patience and love.

I am thankful to my country of origin, Pakistan, for providing me with a free education up to my Bachelor's degree in Electrical Engineering. I am also thankful to my new homeland, USA, for the opportunities I have here.

I am thankful for the guidance and support I received from Dr. Donald Bouldin. With his patient feedback and support, I was able to finish this thesis on time. I would like to thank Dr. Mohammed Ferdjallah for his motivation and encouragement to join the Masters Program in Electrical Engineering. I thank Dr. Gregory Peterson and Dr. Jack S. Lawler for serving on my committee.

I am also thankful to Mr. Mike Galyon for telling me that the best investment I could make was in getting an advanced degree. I also thank Uster Technologies for providing the funding for the hardware and software tools.

I am thankful to my friend, Mr. Arpit Jain, for his support in formatting, reviewing and editing this thesis. I am also thankful to Dr. Roger Riley, Dr. Youe T. Chu, Ms. Bonnie Kerley, Mr. Jonathan Hatcher, Mr. Tony Littleton and Ms. Anjabben Ashraf for reviewing my thesis.

Abstract

The design and implementation of an FPGA (field-programmable gate array) based image analysis system was undertaken to replace an older system whose components have become obsolete. Video from an analog camera is digitized by a video decoder. The data from the video decoder is stored in memory and then processed using an FPGA. The results are then transmitted over a universal serial bus (USB) to a host personal computer for additional processing. The system also controls the timing of a flash to correctly capture the images; it measures color and reflectance and is used to classify the quality of raw cotton by determining the concentration of impurities (e.g. leaves or trash). The original system is first described and the need for upgrading presented. The goals of the new system are then specified and its implementation presented along with the design space tradeoffs that were considered. Finally, the results obtained from using the new system are presented to demonstrate its effectiveness.

Table of contents

List of Figures

List of Acronyms

Acronym	Description
AGP	Accelerated Graphics Port
BGA	Ball Grid Array
CPLD	Complex Programmable Logic Device
DAC	Digital to Analog Converter
DAB	Data Acquisition Board
DDK	Driver Development Kit
DIP	Dual in Line Package
DSP	Digital Signal Processor
EDO	Extended Data Output
EDK	Embedded Development Kit
EAV	End of Active Video
EMI	Electromagnetic interference
FPGA	Field Programmable Gate Array
JTAG	Joint Test Action Group
I2C	Inter-Integrated Circuit
IP	Intellectual Property
ISA	Industry Standard Architecture
LED	Light Emitting Diode
Mb/s	Mega Bits per second
MB/s	Mega Bytes per second
NRZI	Non-Return-to-Zero Inverted

NTSC	National Television System Committee
PCI	Peripheral Component Interconnect
RAM	Random Access Memory
RGB	Red, Green, Blue
RS232	Recommended Standard 232
SAV	Start of Active Video
SCL	Serial Clock
SDA	Serial Data
SRAM	Static Random Access Memory
USB	Universal Serial Bus
USDA	United States Department of Agriculture
VCO	Voltage Controlled Oscillator
VGA	Video Graphics Array
VHDL	VHSIC Hardware Description Language
VIP	Video Input Processor

.

1 INTRODUCTION

1.1 Overview

The measurement of cotton's color and trash impurities and grading it according to color, reflectance, trash contents, and other properties is a big business. In 2006, over 17 million bales of cotton were graded in the US alone. In this country, the United States Department of Agriculture (USDA) is responsible for cotton classing. The USDA currently charges farmers around $1.50 to $1.90 per test. Uster Technologies, Inc. is the major supplier of the automated cotton classing equipment in the US and around the world. The USDA alone has over 500 classing instruments and replaces an instrument on average every eight years. The current color/trash grading instruments use technology developed in the late 90's. Many components in the color-trash classing system are obsolete or very expensive to procure for volume production. There is an urgent need for redesigning and replacing the parts of these systems.

Uster Technologies uses an ISA bus-based image capture card for cotton color and trash area measurements. There are components like the EDA-RAM and video decoder, which have been obsolete for many years and must be acquired through part brokers.

The ISA bus support was discontinued by Microsoft and Intel in the late 90's (around '98-'99) who claimed that the ISA bus created bottlenecks due to its slow speed. Today, almost a decade after that announcement, there are still a few companies using the ISA-based legacy hardware. This is due to a lack of design resources, or in many cases, the original designer has left the company with insufficient information to migrate the design to the next generation buses. Uster Technologies, like a few other companies who still use the ISA bus, pays a premium price to get custom-made ISA bus PCs. In a few years, the likelihood of a PC with an ISA bus will be nonexistent. For these reasons, a redesign of the existing ISA-based image capture card is necessary.

1.2 Goals and Objectives

The intent of this project was to design hardware to replace the ISA image capture and data acquisition card used in the cotton classing instruments. The design should be capable of the following:

a. Field-upgradeable using software that replaces the existing ISA frame capture card on a backward compatible basis,

b. Capturing a composite video signal from the camera and storing the results in a digital format,

c. Flash triggering and synchronization (with the composite video signal) so that the even and odd fields have the same brightness and contrast,

d. Reading the color and reflectance signals (from the voltage-controlled oscillator) by implementing two 16-bit frequency counters for each Color-Head,

e. Converting the captured video fields to a still image frame by converting the stored video data from a BT.656 (YCbCr) format to a RGB format,

f. Capable of transferring the data to a host PC using a USB with a minimum requirement of 12 Mbps but with a preferred 480 Mbps transfer rate,

g. FPGA code that can be updated through a USB port,

h. Last but not least, the software application running on the PC must be capable of the following:

i. Communicating to the hardware through the USB bus,

ii. Initializing the FPGA and the video input processor,

iii. Calibrating the flash timing and color trash module, and

iv. Calculating trash count and area and displaying the image on the screen.

1.3 Optional Goals and Future Tasks

After the completion of the basic design, other optional features that can add value to the design include:

a. Process the image data to calculate trash (count and area) based on the calibration constants, which will be stored in the flash memory,

b. Calculate the color of cotton based on the signal received from the VCO counter and the calibration slope and offset,

c. Provide eight digital proximity sensor inputs and eight high current solenoid driver outputs. This is a very desirable option for the installations where the current system along with data acquisition cards is used to control processes. The availability of general-purpose input and output will eliminate the need for a fully equipped PC and a data acquisition card and save tens of thousands of dollars per installation,

d. Design for manufacturability and testability,

e. Provide flexible technology so that the finished board can be tuned and modified for future requirements with just a download of firmware, and

f. Use of the “Flash-Memory FPGA” for improved security of the code.

2 BACKGROUND

2.1 Introduction

The textile industry is one of the top ten industries in the world. Cotton is one of the most important ingredients of this industry. The United States, China, India and Pakistan are the largest growers of cotton. Cotton is a natural ingredient; therefore, its properties are very inconsistent and are dependent on many factors. To get consistent quality of the products, textile manufacturers measure and closely monitor properties of the incoming raw material. Cotton is graded based on the fiber fineness, length and (tensile) strength of the fibers, its color, and trash contents (leaf, grass, etc.). Classing is the process of measuring different physical properties and grading the cotton based on these properties.

Before cotton can be used in a textile mill, its properties are measured. Mill managers use this data to plan how to mix different bales with different properties to get consistent yarn quality. In the United States, China, and a few other countries, cotton is classed by government funded classing offices, and the data is made available to the buyers. In the rest of the world, the buyer or his agent classes the cotton at private labs to determine the value of the cotton.

2.2 Color and Trash Based Classing

One of the most important properties of ginned cotton is its color and the amount of trash contained in the cotton, mostly in the form of leaf (and some grass). Trash in the ginned cotton mostly consists of cotton leaf, along with some bark and grass, because of automated harvesting. These impurities will have to be removed in the textile mill, because the amount of trash negatively influences the market value of cotton. In most of the world, cotton is hand picked therefore it has very little trash.

There are two main methods available to measure the trash in the cotton. The first one is to mechanically separate the cotton fibers and trash and then weigh the trash. The trash result is displayed as a percent of the weight of the sample. Even though this method is more accurate, it is very time and labor intensive and thus expensive. The other method is to take multiple images of a given sample of cotton. The image is then analyzed to calculate the relative area of trash and number of trash particles in the given surface area of the cotton sample. The result of the test is given in percent of surface area and leaf count. Even though this method is not as accurate as the weight-based analysis, it serves well for the US cotton where labor costs are very high.

Currently there are two major players in the image based trash analysis business worldwide. The first one, Uster Technologies (with over 95% market share) uses an analog camera to capture the image through an ISA-bus based custom image capture card, which is then analyzed by a Windows-based PC. The other company, Premier Polytronics Ltd., uses an off-the-shelf document scanner to scan the cotton sample. Hence, the data is then read and analyzed by a PC. The total hardware cost is lower with Premier's approach (due to the use of a mass-produced scanner), and the resolution is much better than an analog camera. Uster Technologies uses custom hardware due to the slow speed of a line scanner. In addition, a commercial scanner is prone to changes in both its physical and software/firmware. This results in frequent mechanical and software redesigns, which are not only expensive but also a nightmare for those in field service.

2.3 Cotton Color and Trash Measurement System Setup

An analog camera based cotton color and trash (area and parts) measurement system can be divided into three main units:

i) A color, reflectance and image sensor

ii) A frame grabber card

iii) A PC running the application software to acquire and process the data.

Color and trash sensors are marketed as a Color-Head by the Uster Technologies. The Color-Head in itself consists of a color and reflectance sensor module, an analog video camera, Xenon flash lamp power supply and triggering system. The cotton sample is placed on a glass window. The sample is then illuminated by the Xenon flash lamp after receiving the trigger signal from the frame grabber board. The color and reflectance sensor circuit automatically calculates the correct timing on the Xenon flash light curve to sample the cotton. The color and reflectance data is transmitted as the frequency output ranging from 30 kHz to 100 kHz. The camera used in the Color-Head is a standard NTSC camera with a composite video output.

The original design is capable of color video capture, but due to the limited processing power of the PC in the limited time, the image is processed as monochrome. The older frame grabber board [1] is an FPGA based circuit board with 16-bit ISA bus interface. The frame grabber board is responsible for generating the correct timing for the trigger signal for the Xenon flash lamp based on the calibration constant received from the PC. The calibration constant is calculated by continuously triggering the Xenon lamp and capturing the image of a white target. The flash timing is adjusted based on a successive approximation algorithm to make even and odd fields match as closely as possible. This is a one-time process.

A typical color and trash measurement system is shown in Figure 1. The image is captured through the Philips video input processor (VIP) SAA7111. The SAA7111 is a very complex IC with numerous registers to initialize before the image can be acquired correctly. In this project, a new USB based frame grabber board is proposed to replace the existing ISA frame grabber card.

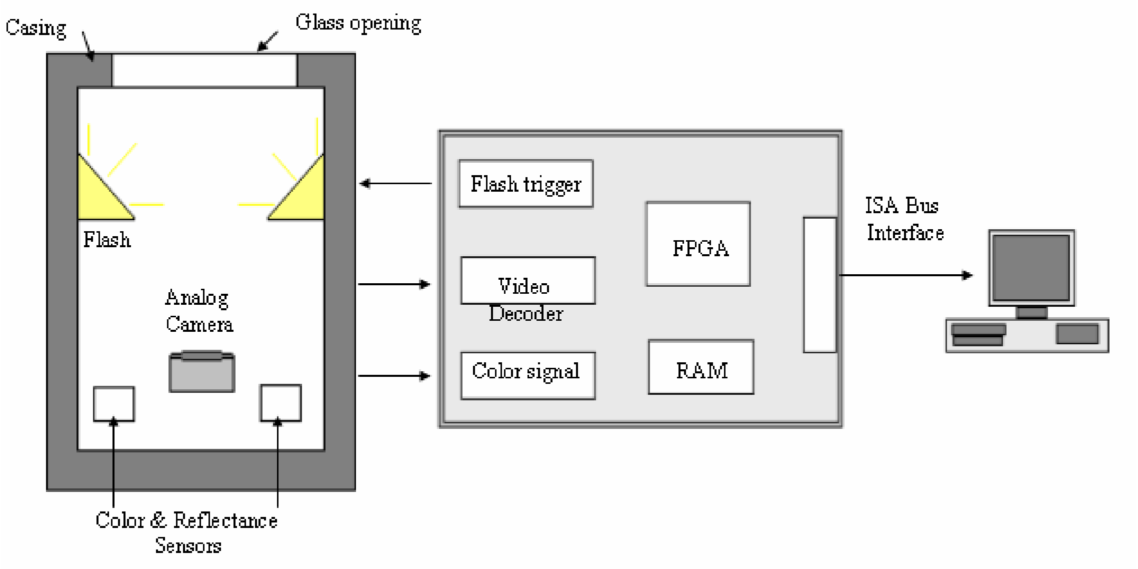


Image acquisition system (also called as Color-Head)	Custom Video Capture Card (ISA bus Based)	Host Computer

Figure 1: Typical Cotton Color and Trash Measurement System

2.4 Motivation

There were several issues with the existing system, which required a redesign. The first was the need to be able to manufacture the video capture cards for the long-term future. With technology rapidly changing, many components on the existing card are no longer manufactured. Therefore, these components are only available from third party part brokers at an inflated price. This puts the manufacturer in a risky situation with uncertain long-term availability of the product. The following are a few other technical and economical reasons to design new hardware.

2.4.1 ISA Bus

The frame grabber uses the ISA (Industry Standard Architecture) bus. IBM designed and introduced the ISA bus with the introduction of the first PC in the early 80’s. It is a 16-bit parallel bus with a very simple protocol. The ISA bus is a very slow bus with maximum throughput of 2-Mbytes/sec. A much faster and more complex PCI bus replaced the ISA bus in the mid 90’s. A 64 bit 66 MHz PCI bus can transfer data at the rate of over 360-Mbytes/sec. For video applications, a new standard AGP (Accelerated Graphics Port) bus was introduced in the late 90’s. The latest standard for the plug in cards is a PCI-express bus (which is a serial bus). Since the ISA bus has long been obsolete, it is very difficult and expensive to purchase the ISA bus based PCs. Only two or three companies are offering any ISA bus based PCs and they cost about $800 more than an off-the-shelf PC. This is the main reason to start the redesign effort.

2.4.2 Slow Data Transfer Speed

Currently the system is capable of capturing and transferring slightly more than two full color or six monochrome images at 640 X 480-pixel resolution. It has been decided to use mega pixel cameras in the future and to capture multiple frames per second from each of the two Color-Heads. The theoretical maximum speed for the ISA bus is 2-Mbytes/sec, which is much slower than modern buses like USB 2.0 (480 Mbps around 48-Mbytes/sec).

Uster Technologies has some other data acquisition and processing cards that are based on the ISA bus. Therefore, the desirable outcome would be to implement this board on a flexible platform, so that other boards can also be upgraded to the USB bus.

2.4.3 PC Dependent

The existing system is very bulky, complex, and expensive. For a simple color trash measurement, you have to have a full-blown PC. In many applications, where a customer has multiple Color-Heads distributed around the facility, a dedicated PC (with a monitor) is provided for each Color-Head. The whole system can be greatly simplified by connecting the frame grabber to a cheap microcontroller board. Many of the microcontroller-based systems can be connected to the main PC via wireless network. This will save tens of thousands of dollars at each site.

2.4.4 No Processing Power

The current sensor is not a self-contained “Smart Sensor”, i.e. the current design does not carry intelligence to process the image data and directly produce results. It does not utilize the fast processing of the FPGA. A further evolution to the previous point may be to use the FPGA to do the image processing. This will save around $1000 per installation (in PC, monitor and operating system costs) and will make the system much more compact and reliable.

3 SYSTEM DESIGN

3.1 Introduction

Before starting hardware implementation of the system, the top-level system block diagram was drawn to identify the essential components/blocks of an image capture system. As Figure 2 shows, there are four main components of the color trash processing hardware:

a) An analog video decoder, which converts the incoming analog video stream into digital video stream.

b) A control and processing unit which decodes the incoming digital video stream, generates the proper memory address and control signals, and writes that data into SRAM (or DRAM). This unit also performs many other essential but less critical tasks like initializing the video processor through an I^2C bus, generating flash timing signals, reading the data from the memory, and eventually sending it to the USB controller when demanded by the external host (PC).

c) Memory, where the video data is stored temporarily before it is read back. The temporary storage is necessary for the data to be read at an extremely high rate (27 M samples/sec.) from the video input processor. The USB controller and the PC cannot handle that.

d) Communication interface between the PC and the Controller. The communication interface can be implemented (or be part of) the control unit or it may be an external piece of hardware.

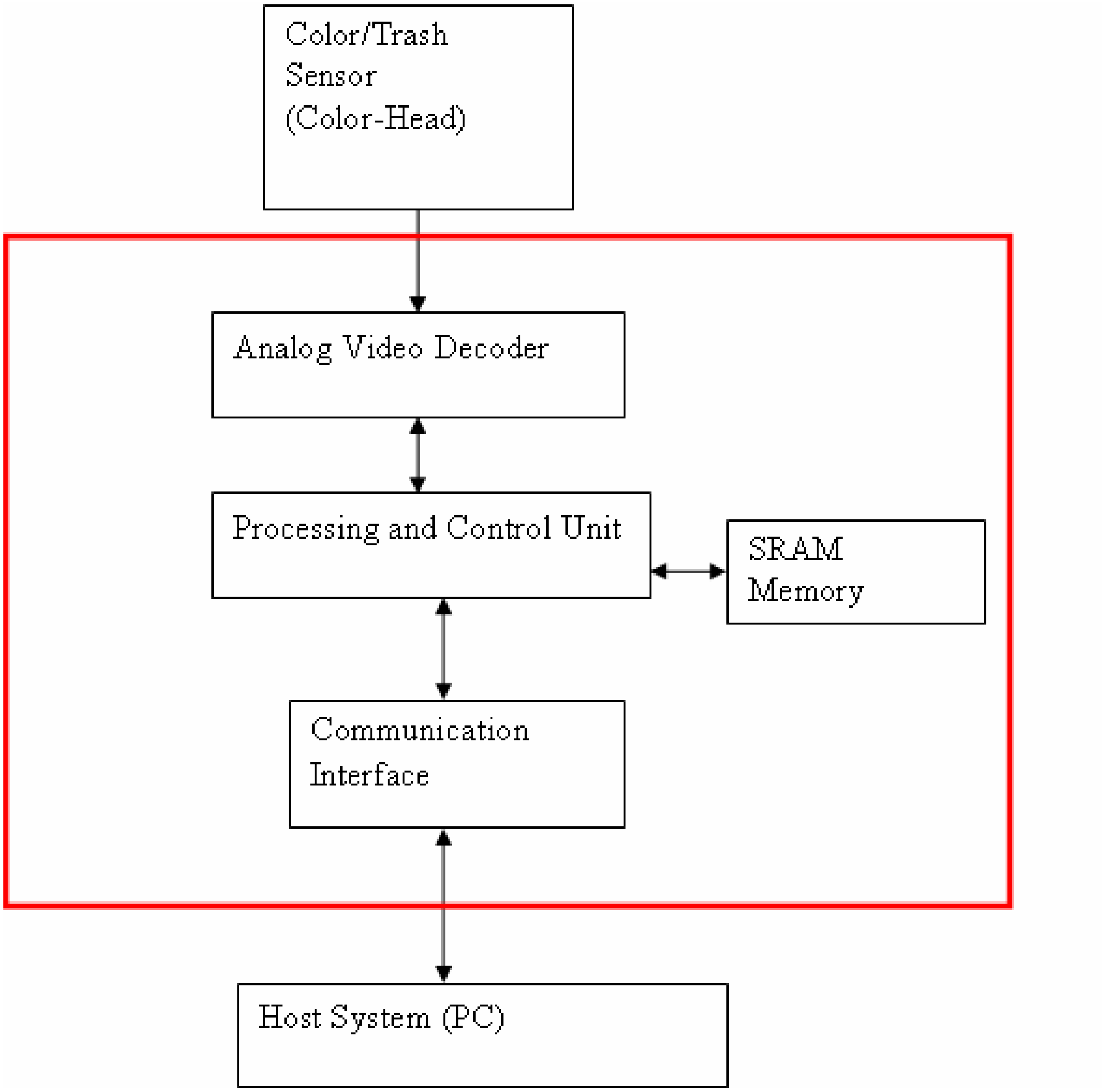


Figure 2 : Color Trash Processing Hardware

3.2 Technologies and Components Selection

Once the different building blocks of the design were identified, the next step in the evolution was to identify and select the components and technologies best suited for each block. The following is the justification as to why each component was selected. The final design was meant to be manufactured in large quantities, and there was a fixed amount of time to finish the project. There were a few guidelines that were followed during the project.

3.3 Criteria for the Selection of Components and Technologies

The selection of different components and technologies were based on the following well-defined criterion:

1. Design time, which includes the time to learn the development tools associated with the technology or components; for example, learning a new language (Verilog or Assembly language).
2. Availability of a functional reference design set so that the development process could be jumpstarted. Otherwise, one would have to design a board in parallel while the code was being written so it could be tested. In a custom-designed board, there is higher risk of going through many reruns. In addition, if there is an issue, it is difficult to figure out whether it is due to hardware or software.
3. Support for the USB driver development for Windows XP to save on driver development since this is a considerable investment of time as the programmers are not familiar with driver development.

4. Flexibility to reuse some of the design components to upgrade other designs (Uster Technologies has many other ISA bus DAB and DSP boards in other instruments).
5. Bill of material cost, even though it is not a big issue due to low production volume, but it is still desirable to keep the product cost low without compromising the reliability.
6. Low cost of development tools (software) is desirable since the goal is to keep the total design and development cost below $6000.

3.4 Choice of Processing and Control Unit

For the processing and control unit in the new frame grabber development, the following were the choices in technologies:

1. An FPGA-based implementation, with either soft USB core (IP) or an external USB controller. Xilinx, Altera and Actel are the three leading suppliers of FPGA based technology.
2. A microcontroller or microprocessor-based design, with on-chip or external USB controller, i.e. ARM, Freescale, or Silicone Labs' microcontrollers.
3. Using a digital signal processor like Texas Instrument's DSP (with built in USB support).

3.5 Note on Comparison Between Technologies

In terms of time, an FPGA-based hardware design will take longer to complete than a microcontroller or a DSP-based design, but the FPGA is the most flexible and is the most powerful technology. A USB-based microcontroller is not fast enough for the

720 X 480 analog video capture. The analog video signal is decoded into an 8-bit digital signal at a rate of 27MHz. Therefore, for a microcontroller or DSP to capture a frame, incoming data must be processed much faster than the incoming information to decode it for information like START and END-OF-LINE, field, and frames. Once the microcontroller or processor has decoded the signal, it then can write that data to the RAM to be retrieved later and send it to the PC. A quick survey of the market has proven that the only other technology other than FPGAs capable of performing such tasks were the high-end DSPs such as TI's 700MHz DSP. These DSPs were not only expensive but were also difficult to design, as the learning curve was too steep. On the other hand, FPGAs are ideally suited for this kind of application. They are super fast because of the ability to perform multiple operations in parallel. The FPGA is also very flexible which is important because the baseline design can be used to upgrade other hardware in different products that are ISA bus based. Another advantage of using an FPGA-based design is that more horsepower is available which may be useful if there is a decision to process all the data on-board. The FPGA can perform image analysis (calculate count and area) 10-100 times faster than a microprocessor due to concurrent processing.

Once the decision to use the FPGA was made, the choice was between three or four manufacturers. When choosing between different manufacturers, a comparison was made of the different vendors based on the following:

i. Availability of a video capture evaluation platform with most of what was needed, like SRAM, USB, and Ethernet. The example code from the manufacturer or distributor in VHDL or Verilog was very desirable.

ii. Availability of devices in non-BGA packages as BGA is difficult to work with.

iii. Soft-core processor desired, like NIOS II or Micro-Blaze/Pico-Blaze.

iv. Cost per chip.

3.6 Choice of Communication Interface

There were three commercially available, high-speed standards to choose from for the communication interface bus. This interface was to be the medium of communications between the controller and the host CPU on the PC. The three most common high-speed serial interfaces, USB, Ethernet, and IEEE1394 were compared. Looking at Table 1, the USB bus is a clear choice, as it has the highest favorable score. For implementing the USB interface, three choices were available:

1. A USB soft-core from Synopsys (or other supplier) implemented in the FPGA.
2. A dedicated USB controller.
3. An RS232 bus to a USB converter IC.

Even though the RS232-to-USB converter is the fastest route to complete any USB-based design, it was lacking bandwidth. For an image capture application, the maximum RS232 baud rate is not high enough and it was the limiting factor. The USB IP or soft-core was not chosen, as Synopsys's marketing indicated that it was best suited for very high volume applications. There were other USB cores available from other sources, but good documentation was lacking for the PC software and driver development. This indicated a high future risk when a new operating system is released.

Table 1 : Communication Interface Choices

Advantages	**Disadvantages**
1. Ethernet	
a. Long range of operation (328 feet). b. Commonly used protocol. c. It is supported by most modern computers.	a. Generally used for network of computers and not as a single communication channel.
2. IEEE 1394	
a. High speed (1394b is 800Mbps). b. Specifically used for video signals. c. Ease of connection. d. Hot pluggable.	a. Not supported by older computers. b. Not widely used in industry. c. It has short range of operation.
3. USB 1.1 and 2.0	
a. High speed (up to 480 Mbps). b. Availability of multiple ports on PC. c. Simple connections. d. Range up to 5 meters. e. Easily available drivers. f. Hot pluggable.	a. Very complex protocol. b. Hard to program firmware and software.

After careful consideration of the above factors, the decision was made to use a dedicated USB controller. Then a long list of suppliers was available to choose from. The decision was made in favor of Cypress after consulting other engineers and browsing through USB texts, based on the following criterion:

i. Ease of implementation.

ii. Available example code.

iii. Evaluation and programming tools available from the manufacturer and third party suppliers.

3.7 Selection of Video Decoder Processor

There are many video decoders available from multiple manufacturers that have more or less the same specifications. The decision was made in favor of the Philips SAA7113 video input processor, because the legacy board used the Philips SAA7111 (which is now obsolete). This video input processor (SAA7113) is very similar in functionality to its predecessor. Therefore, by staying with the same supplier it was ensured that the image data would be consistent.

3.8 Selection of Memory

The memory choice was between SRAM and DRAM. SRAM is much simpler to interface compared to DRAM. DRAM is available in much higher densities at a significantly lower cost. The decision between SRAM and DRAM was left to the development board. In other words, it was decided to use whatever memory the development platform designer used.

3.9 Development Platform Choices

Once the decision was made to use an FPGA, the Philips video input processor and the Cypress USB controller; a search was started for an FPGA development board with those devices. The following is a comparative survey of what the closest matches were.

3.9.1 Altera DE2 Education and Development Board

The Altera DE2 development system shown in the Figure 3 was the first choice. It has the following features:

a. Cyclone II FPGA.

b. Nios II soft-core processor.

c. Philips video input processor.

d. USB host and device support.

e. Single board with all capabilities.

f. Good documentation and demonstration examples.

The drawbacks of the Altera evaluation board for this project were:

a. The Philips USB 1.0 controller was on the board but the commonly used Cypress EZ-FX2 was desired.

b. The demonstration and example codes were written in Verilog but VHDL was preferred due to past experience.

c. Learning of new EDK, QuartusII, would be required.

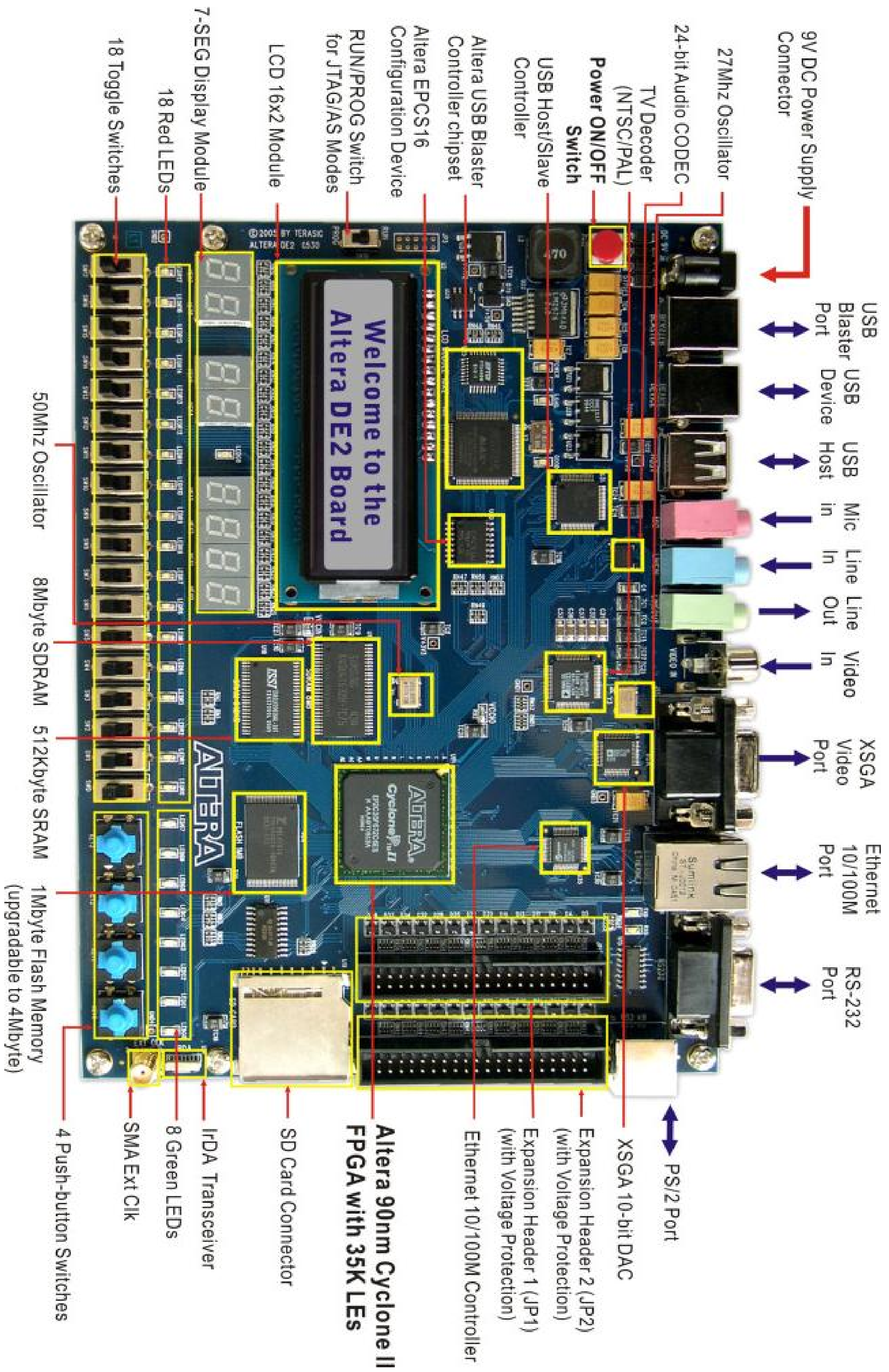

9V DC Power Supply Connector
27Mhz Oscillator
24-bit Audio CODEC
TV Decoder (NTSC/PAL)
Power ON/OFF Switch
USB Host/Slave Controller
Altera USB Blaster Controller chipset
Altera EPCS16 Configuration Device
RUN/PROG Switch for JTAG/AS Modes
LCD 16x2 Module
7-SEG Display Module
18 Red LEDs
18 Toggle Switches
USB Blaster Port
USB Device
USB Host
Mic in
Line In
Line Out
Video In
XSGA Video Port
Ethernet 10/100M Port
RS-232 Port
PS/2 Port
XSGA 10-bit DAC
Expansion Header 2 (JP2) (with Voltage Protection)
Expansion Header 1 (JP1) (with Voltage Protection)
Ethernet 10/100M Controller
Altera 90nm Cyclone II FPGA with 35K LEs
SD Card Connector
IrDA Transceiver
8 Green LEDs
SMA Ext Clk
4 Push-button Switches
50Mhz Oscillator
8Mbyte SDRAM
512Kbyte SRAM
1Mbyte Flash Memory (upgradable to 4Mbyte)
Welcome to the Altera DE2 Board

Figure 3 : Altera Development Board

3.9.2 Digilent VP4 Development Board with Digilent Video Decoder Board

The Digilent VP4 development system was the second choice due to good experience using Digilent products. It has the following advantages:

a. The Virtex-4-Pro is a high-end FPGA from Xilinx.

b. Good documentation and support from the vendor.

c. All desired devices were present on the board.

The drawbacks of the Digilent VP4 evaluation board for this project were:

a. The Digilent video decoder board is based on Analog Devices' ADV7183 video decoder.

b. The Virtex-4-Pro FPGA is costly and is only available in BGA packages.

c. A two-piece solution consisting of main and an A/V card is undesirable.

3.9.3 Avnet – Xilinx Evaluation Board with Avnet A/V Card

The Avnet – Xilinx Spartan3 Evaluation system was the last choice due to the very high cost of the development system. The Xilinx Spartan3 board is shown in Figure 4. Avnet's Audio-Video board is shown in Figure 5 and the block diagram in Figure 6. The two-board system has the following advantages:

a. USB 2.0 support with desired Cypress controller (CY7C68013).

b. Optimal sized Xilinx Spartan3E FPGA.

c. On-site vendor support.

d. Multiple video inputs and outputs (SVGA, etc.).

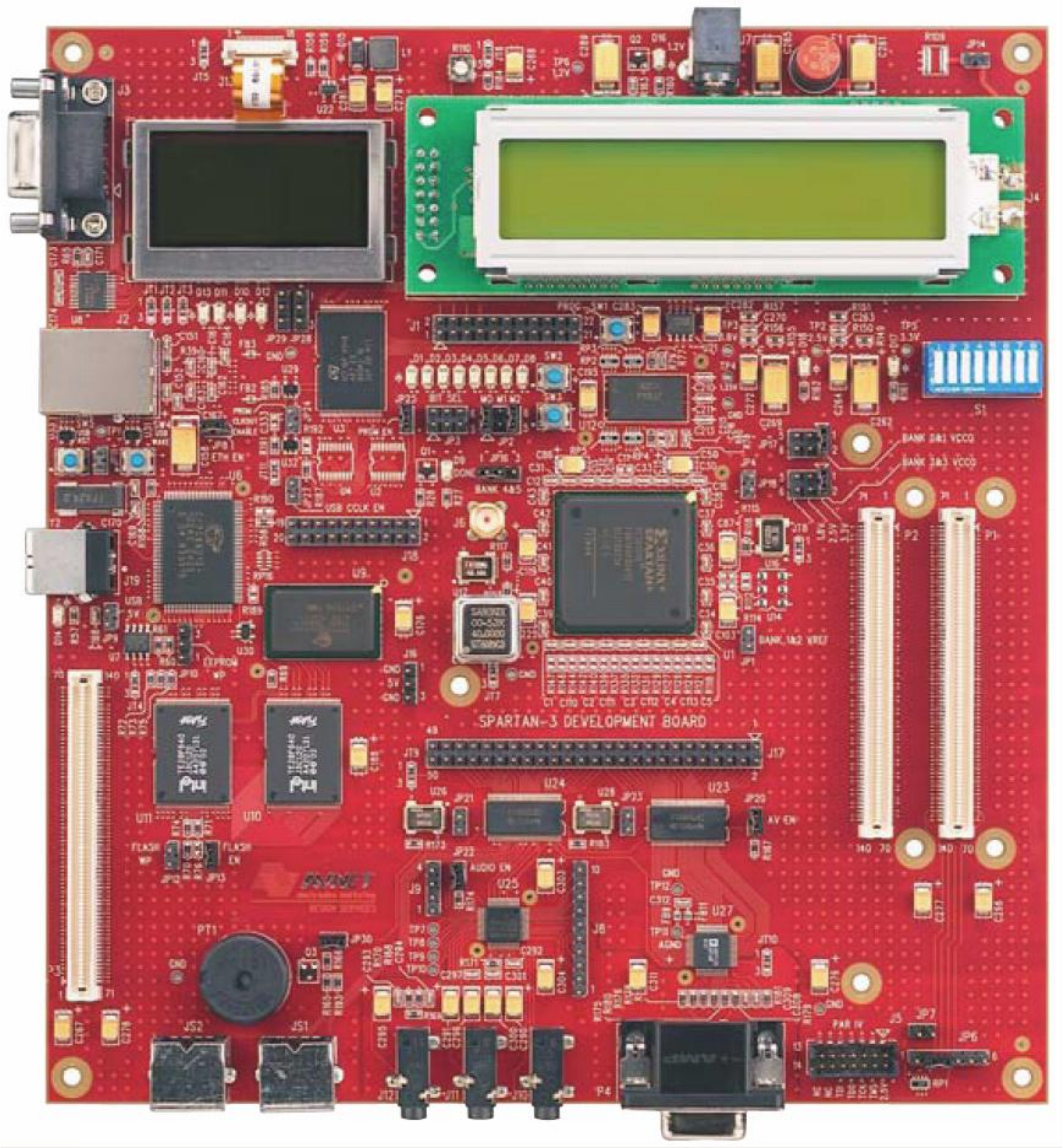


Figure 4 : Avnet’s Xilinx Spartan3 Development Board

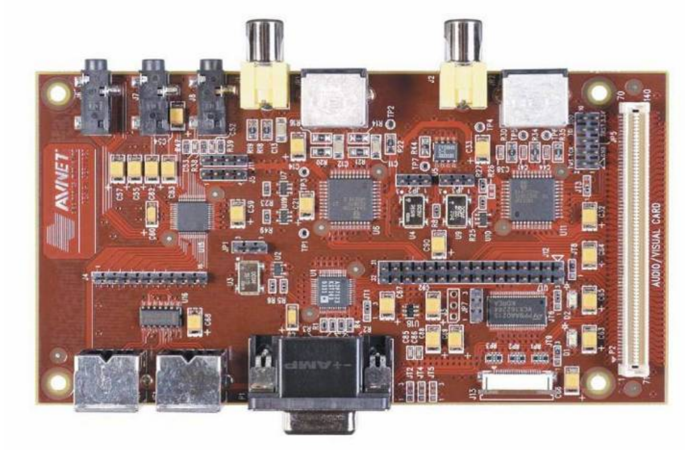


Figure 5 : Avnet Audio Video Development Module

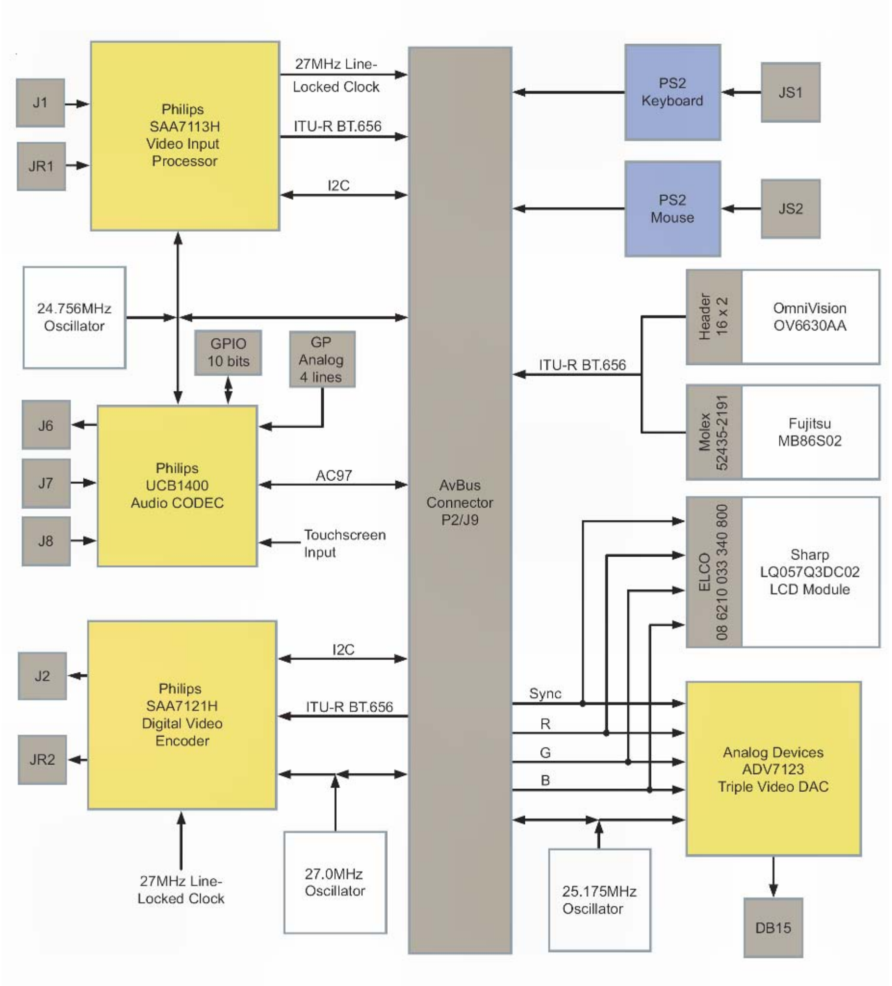


Figure 6 : Avnet Audio Video Development Module Block Diagram

e. Demo code in VHDL.

f. The video processor SAA7113 is preferred due to past experience with SAA7111.

g. Supports familiar Xilinx ISE development EDK.

The drawbacks of the Avnet evaluation boards for this project were:

a. A two-piece solution consisting of main and A/V cards is undesirable.

b. No demonstration and example codes for either the video capture or the USB were available.

c. High hardware cost ($700 + $300).

d. The board requires a lot of initial setup and is not operational off-the-shelf.

e. A programming cable and support software is not provided.

3.10 Selection of Development Platform

Between Xilinx and Altera, the choice was made initially in favor of Altera based on the availability of the evaluation board. Altera had an evaluation board with a video decoder on board which could save thousands of dollars in prototyping costs and the code could be written right away. The Altera DE2 evaluation-board cost is low and it is well documented and comes with some example codes. The only major issue was that the USB interface is based on an older Philips chip with a maximum bit rate of 12Mb/sec. There were two other minor issues: one being that all the code was in Verilog and the other being the learning curve associated with the new development environment (Quartus II).

The second choice was the Xilinx Spartan3E development board from Avnet with an older audio video board. This two-piece solution was equipped with all of the necessary hardware that was needed to develop a fully functional frame grabber card. It has a video decoder and an encoder as shown in the block diagram in Figure 6. It also has an FPGA big enough to capture, decode and process the video and a high-speed (480Mb/sec) USB processor. The only drawback was that there was no example code for the video capture (decoder) and no support for the custom USB driver development.

3.11 Finalizing Major Components of the System

a. The FPGA as the central processing and controlling unit – Xilinx Spartan 3E.

b. A USB bus controller – Cypress CY7C68013A – EZ FX2 USB controller.

c. A video decoder for conversion of component video – Philips SAA7113H.

d. A flash triggering signal generator.

e. A signal conditioner (Schmitt trigger) for input color and reflectance signals.

f. General-purpose input/outputs - GPIOs for actuators and relays.

g. An SRAM for storing the image data.

h. A flash PROM for storing the FPGA configuration file, Intel flash XCF16P.

i. Host-side software for control and communication with the frame capture board to be written in Visual C++ and Microsoft DDK.

4 IMPLEMENTATION AND RESULTS

4.1 Introduction

Once the components and evaluation hardware was selected, the next step was to start writing the VHDL (and C) code for the design. Before writing the code, the system block diagram needed to be defined and the interconnection between different components identified. Afterwards, the design flow was defined by prioritizing different tasks in a logical sequence. The TOP Level block diagram of the complete color and trash measurement system is shown in Figure 7. The flow of the data is from the video decoder and color sensor to the FPGA. There is a bidirectional data transfer between the FPGA and the memory, as well as between the FPGA and the USB Controller. The FPGA also controls the Xenon flash lamp trigger timing and initializes the video input processor through the I^2C bus. There is a bidirectional data flow between the USB controller and the host PC. It is evident from the block diagram that the FPGA has the most important position in the design, as it controls all the data flow between every other device. The following is a brief summary of the tasks the FPGA performs and the steps it takes to capture a single video frame and send it to the PC through the USB Bus:

a. Reset all the devices after power ON event.

b. Initialize the video input processor and video decoder using the I^2C serial interface.

c. Decode the digital video data from the video input processor and stay synchronized to the video.

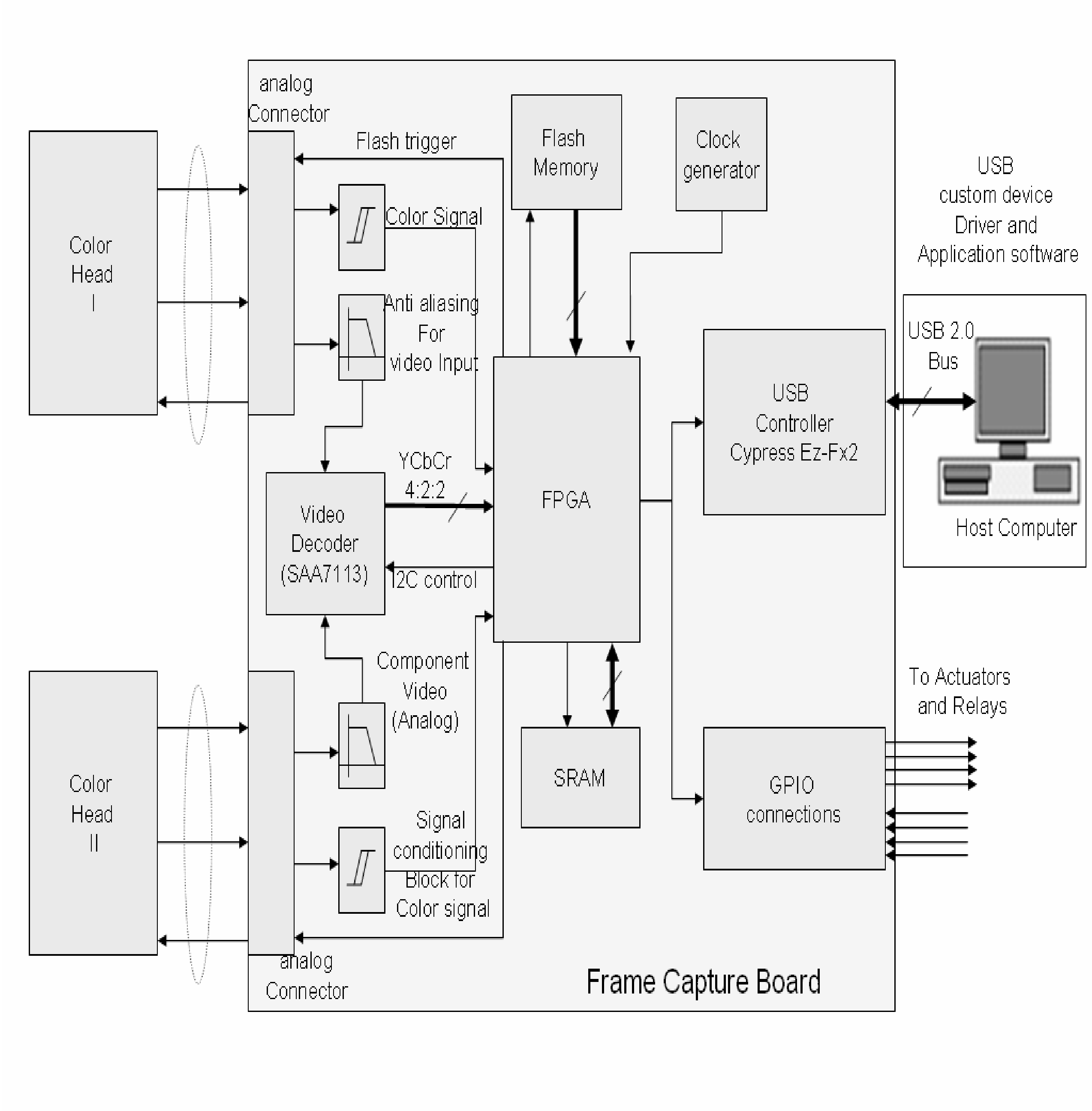


Figure 7 : Block Diagram for the Cotton Color Trash Measurement System

d. Automatically generate appropriate memory addresses for all 1440 X 480 bytes while staying synchronized with the video signal. The top left corner pixel is assigned address 0. The addresses increment by one towards the right and 1440 per line towards the bottom of the screen.
e. The FPGA generates the correct trigger timing for the Xenon flash lamp, synchronized with analog video and based on the given delay.
f. After the Xenon lamp is flashed, the FPGA writes the digital video data into SRAM based on the synchronous addresses generated by it.
g. Reads data from SRAM and sends it to the USB controller when requested by the USB controller.
h. Wait for the next trigger and repeat the cycle.

4.2 FPGA Functional Partitioning

The above list is by no means a comprehensive list as it does not include exception handling and diagnostics (and many other functions), but it serves as a starting point. To accomplish the above tasks in an organized manner, the FPGA is partitioned into many small modules (see Figure 8), with each module performing a specific task. The following is a list of the main modules required for the video capture and transfer to the PC:

a. The Main module is the top-level state machine which handles the interactions between different modules, external hardware and data flow.
b. The I^2C module converts the parallel data into serial data according to the I^2C standard specifications.

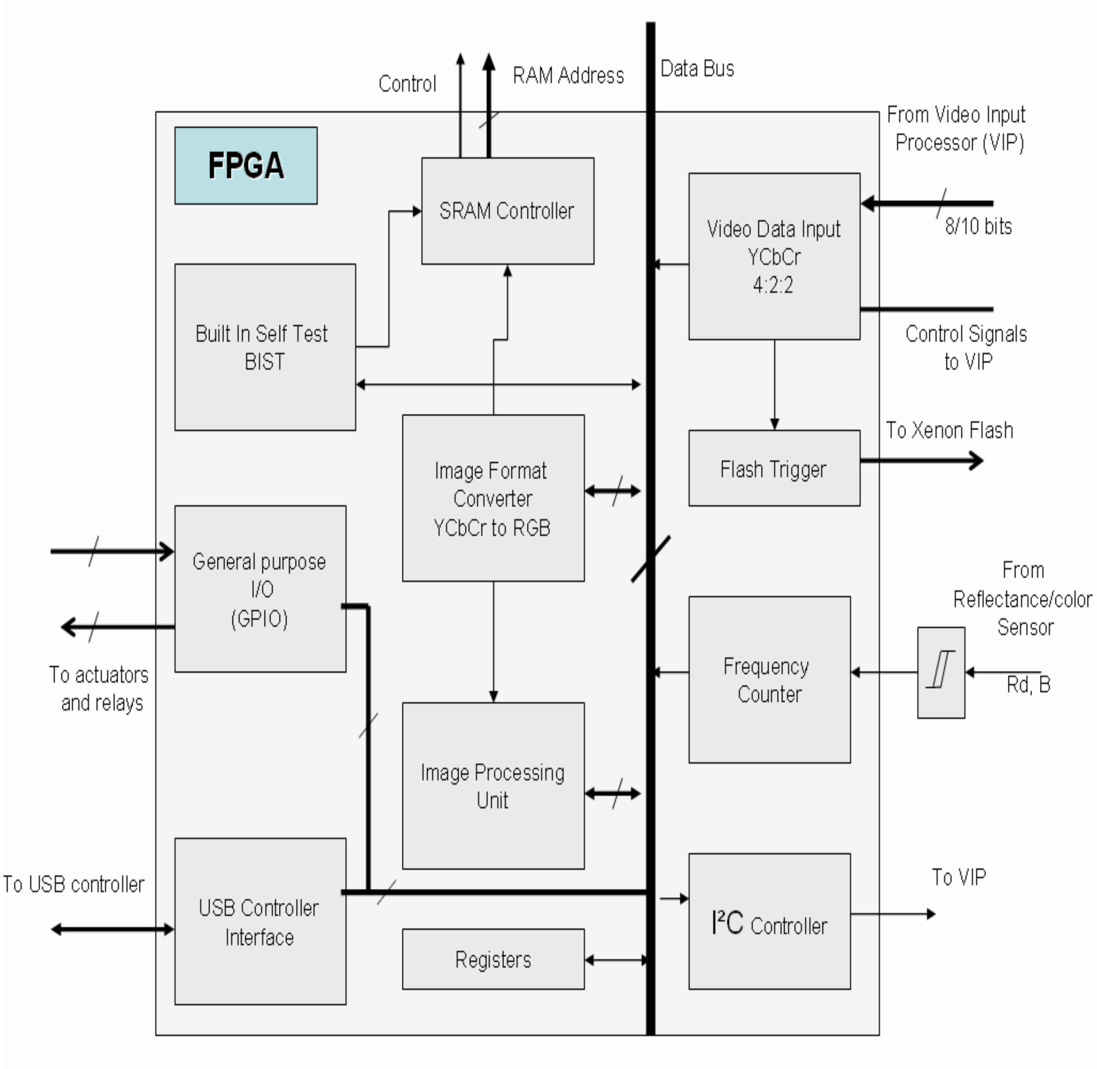


Figure 8 : Block Diagram for the FPGA

c. The video processing unit decodes the incoming video signal and generates the addresses and the sync signals for the other modules to synchronize to the video signal.

d. The flash trigger timing module generates the flash trigger signal.

e. The SRAM interface module generates the appropriate logic and timing signals to interface the SRAM memory. It also performs the logical conversion so that the 512K X 32-bit wide SRAM looks like a 2M X 8-bit linear memory space to all modules.

f. The USB controller interface handles all the timing and logic to transfer data between the SRAM and the USB controller as well as between the SRAM and the FPGA.

g. The video pattern module generates the video pattern for the diagnostics to verify the design of different modules.

4.3 Hardware Setup and Testing

As mentioned in the previous section, the development hardware consisted of two boards. The first was the Xilinx FPGA development board with a USB and parallel interface. The second was a video development board with the parallel interface compatible to the FPGA board. These boards were not designed or marketed for each other. Therefore, no test code existed (at that time) to connect these boards together and test them.

The video encoder board and the FPGA development board shared Avnet's proprietary parallel bus interface. After comparing the schematics, it was possible to connect the video board to the FPGA board.

The development system for the FPGA was tested first by writing a simple VHDL code to turn some LEDs ON and to read the input switches and push-buttons. This simple task was not without challenges due to a device programmer incompatibility. After ordering a new low voltage JTAG device programmer, the device was programmed and worked as expected.

The video development kit only came with the code to generate color bars on a VGA monitor but no user configuration file. The VGA bars were generated on-the-fly (no buffer was used). This code was written for an older Virtex FPGA board and an older audio video board (in 2002). This code was modified to do the same on this setup (Spartan3 development kit and Avnet audio/video board). This code was the starting point for the rest of the VHDL code. The large number of extra hardware (components) on the boards compensated where the Avnet evaluation tools were lacking in the example code. There were two extra components (the video encoder and the video DAC) which were not part of the final design but were very useful as diagnostics tools.

The audio/video development board has three main video components:

1. The VIP or video input processor which digitizes the input video signals and converts it into BT.656 digital video.
2. The video encoder which takes the digital video information in BT.656 format and converts it into a composite video signal.
3. The VGA DAC for displaying data on a standard PC monitor.

4.4 VGA Color Bars Generator

The VGA color bars generator code was straightforward. It changed the color value after every 64 pixels (on the horizontal line), and this pattern repeated in every line. Each pixel was given a value based on the horizontal address value. Since the pixel values are repeated throughout the scan (line after line), it results in color bars. This code was modified to generate different patterns on the VGA screen where the color value was the function of the X-Y position. By using different combinations of address dependencies, different patterns were produced on the screen as shown in Figure 9.

Once this video color bar code was working, it was verified that the connection from the Xilinx board to the AV board was correct. It was also verified that the FPGA to video DAC interface was working properly. Even though a video DAC is not needed in the final design, it was realized early on that a video DAC could be used as a very powerful debugging tool.

4.4.1 Video Monitor as Debugging Tool

The objective of this design was to capture “analog video” after generating a “flash trigger pulse”. The frame captured must have two (even and odd) fields of equal brightness. The above objective can only be accomplished by pre-triggering the “Xenon flash" in such a way that the “flash” occurs in the middle of the blanking pulse. The captured frame (720 x 480 pixels) is to be stored in RAM and then sent to the PC through the USB interface. From the camera, the signal travels through the video input processor to the Xilinx FPGA to SRAM, from SRAM back to the FPGA and from the FPGA to the USB-Microcontroller and to the PC as shown in Figure 10.

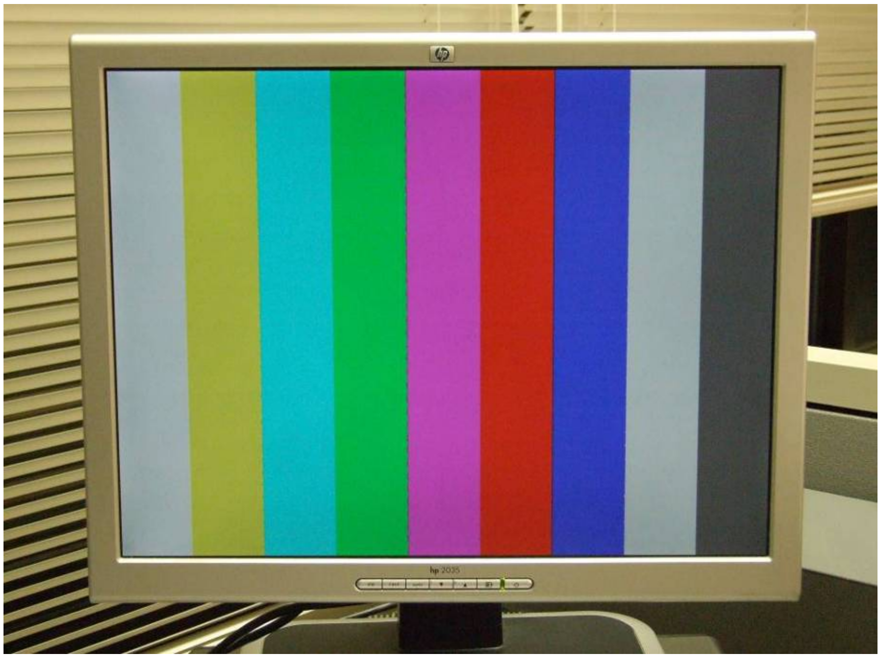


Figure 9 : Color Patterns

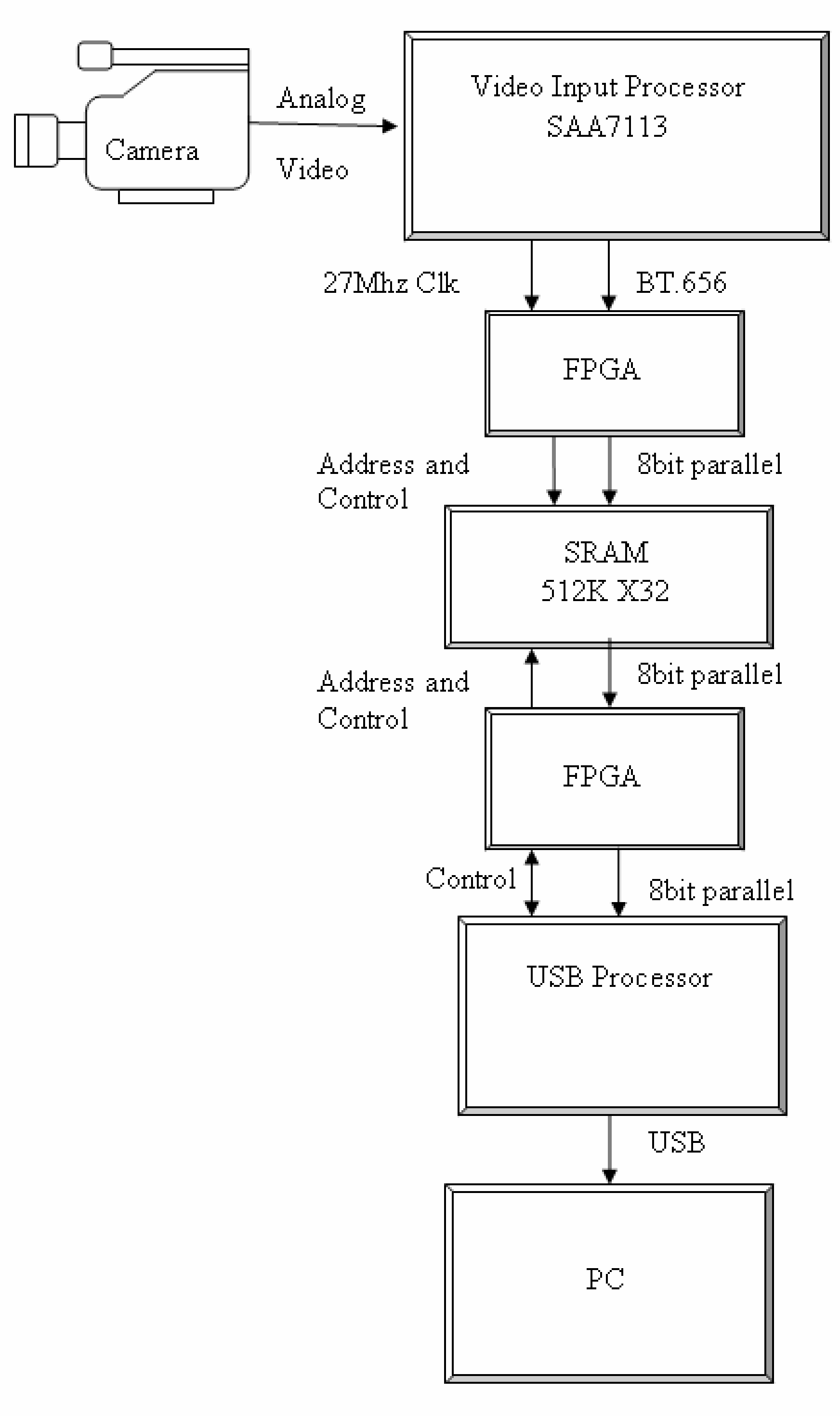


Figure 10 : Data Flow from Camera to the PC

As Figure 10 shows, the data flow is very complex. This system would be very difficult to design and debug completely in a single try. Even though there are many modules in the FPGA design, the most risky part of the entire design is the USB interface, due to the lack of the experience in USB design.

To make the design job manageable, it was divided into smaller sub-designs, where each smaller design (module) can be individually tested and verified. It was determined that a VGA monitor could be used to display the video data before writing it to the RAM.

Filling the SRAM with a known pattern and then displaying that pattern on the VGA screen verified that the state machine that reads and writes to the SRAM was functional. The pattern generator routine was tested by displaying the pattern on the VGA screen. The same technique was used to test the USB hardware by generating a known pattern on-the-fly for the USB data and then displaying it on the PC.

4.4.2 Initialization of Video Input Processor

Once the FPGA board and the video decoder board were tested and verified by generating the vertical bars, the next step was to initialize the heart of the analog video capture card, the "video input processor" (or VIP). The video input processor was capable of converting analog video from many different formats and interfaces into digital video format. Before the video input processor can be used in any application, it needs to be initialized through the I^2C bus. Since a VIP covers many different standards, there are many registers to be understood and initialized. A long list of variables in the decoder determines such things as video format, gain, timing, filtering, and many other parameters. These values are initialized by the FPGA through the I^{2}C bus. Since the FPGA does not have an I^{2}C communication interface built-in, an I^{2}C communication module must be designed. This communication module must read the values of the different registers (over 50 bytes) and serially send these bytes to the corresponding address. Before describing the design of the I^{2}C interface, the following section defines the I^{2}C bus interface and communication protocol.

4.4.3 I^{2}C bus

The I^{2}C is a low-cost and low-speed serial communication bus. The name I^{2}C stands for inter-integrated circuit and is pronounced “I-Squared-C”. The I^{2}C bus was invented by Philips to be used on home entertainment equipment. The I^{2}C is a multi-master serial communication interface where all the devices are connected and communicate with each other through only two wires: a serial data line (SDA) and a serial clock line (SCL). Both of these bus lines on the I^{2}C bus are open collector (or drain) type with pull-up resistors on the bus. Since all the devices communicate through only two wires, the PCB design is simplified by the use of an I^{2}C bus. The original I^{2}C standard defined the bus voltage to be 5V and the maximum communication speed to be 100 kHz. Later revisions allowed other voltages (3.3V) and speeds up to 3.4 Mb/sec.

Each device on the I^{2}C bus is recognized by its unique address and can be operated as either a transmitter or receiver depending on the function. In addition, each device can be considered as either a master or slave when performing data transfer. A master is a device that initiates the data transfer and generates the clock signals for that transfer. There can be more than one bus master as an I^{2}C is a multi-master bus, but when one master initiates the transfer all other devices on the bus are considered as slaves.

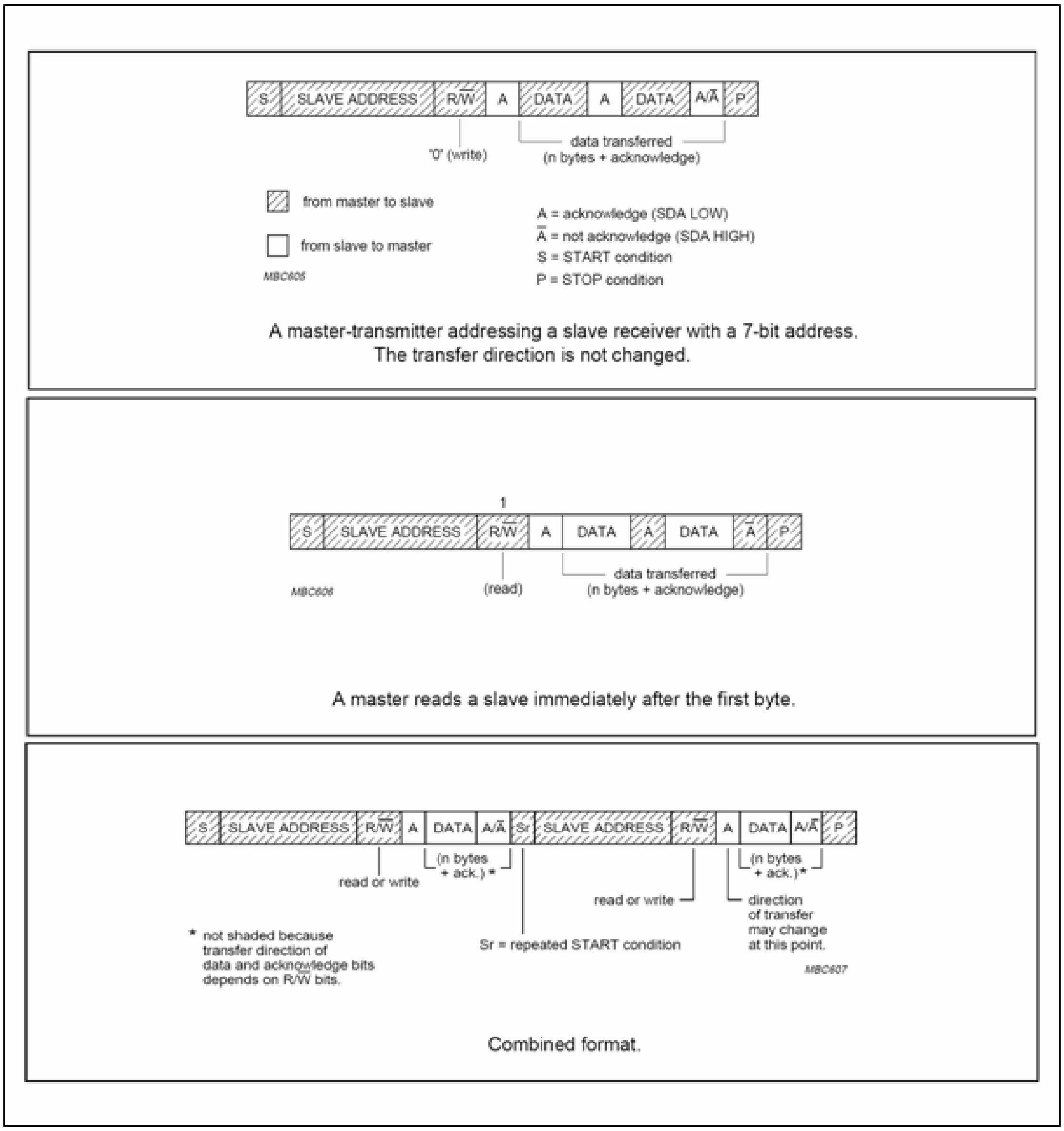
There are three possible data transfer modes on the I^2C bus as illustrated in Figure 11:

1. The master transmitter and the slave receiver.
2. The master reads the slave transmitter after transmitting the first byte.
3. A combined format where the data transfer direction changes during the transfer.

Both SCL and SDA are bidirectional lines with pull-up resistors connected to the positive power supply. When the bus is free, both the lines are high. The outputs of the I^2C buses are open drain and pull the line low (or leave high) in order to communicate. If more than one master tries to initiate a data transfer at the same time, I^2C specs provide an arbitration procedure that ensures that only one device is allowed to control the bus and the data is not corrupted.

Data transfers are initiated with the START condition and are terminated with a STOP condition as shown in Figure 12. Normal data stays stable during the high period of the SCL and only changes when SCL is low. START and STOP conditions are unique cases where SDA changes when SCL is high. When SDA changes from high to low while SCL is high, it is a START condition. Similarly, when SDA changes back to high (from low) when SCL is high, it is a STOP condition.

Data is transferred as 8-bits, followed by one acknowledge bit, so each byte transfer takes 9-bits. Standard communication on the bus between the master and the slave is composed of four parts: START, slave address, data transfer and STOP. The I^2C standard allows both 7-bit and 10-bit slave addressing. In 7-bit addressing, the bit followed by a 7-bit address is a read/write bit. If the read/write bit is "1", it indicates a read operation while a "0" indicates data



A master-transmitter addressing a slave receiver with a 7-bit address.
The transfer direction is not changed.

A master reads a slave immediately after the first byte.

Combined format.

Figure 11 : The Three Data Transfer Modes

(Source: I^{2}C standard specification January 2000, pages 14 and 15)

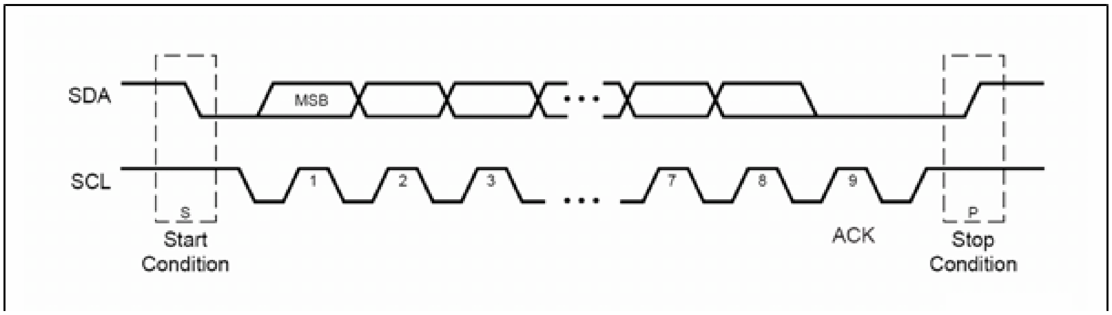


Figure 12 : Data Transfer on the I^{2}C Bus

(Source: Xilinx Application Note XAPP385)

transmission or a write-to-slave operation. When the slave device is addressed and receives its address, it acknowledges by pulling the SDA line low on the ninth clock. After the master has received the acknowledgement (ACK) from the slave device, it then transmits data byte by byte. The master device terminates the transfer by generating the STOP signal.

For the I^{2}C controller, many options were considered. The first option was to write a VHDL state machine for the controller. After some research and studying the I^{2}C standard, that idea was dropped in favor of acquiring the I^{2}C core. The first choice was the Xilinx I^{2}C bus controller as defined in Xilinx application note XAPP385. This core is available free of royalty as VHDL source code. The Xilinx core design was very well documented. The only issue with that core is that it was designed to be used with the microcontroller or a microprocessor and be implemented on a small CPLD. Since this core was designed for a small CPLD, the size was not an issue but the lack of the intelligence in the FPGA was. Since the frame capture design only required one-way data traffic from the I^{2}C bus, adding this more sophisticated core meant adding complexity to the main/top-level controller. There were other cores available on the Internet but almost all of them were designed to add I^2C capability to the microcontroller. Avnet's technical support engineer provided the I^2C initialization code, which was adopted for this design. This code was written for another very similar video input processor, also from Philips (now NXP). This core worked flawlessly with this design after the initializing values were modified for the SAA7113 (video decoder/processor and SAA7121H Philips digital video encoder).

4.5 Video Sync Unit

Once the video encoder SAA7121 and SAA7113 (VIP) were initialized, the FPGA started receiving the video data stream. The digital data routed to the digital video encoder from the FPGA. By sending the digital data to the video encoder (SAA7121), it was verified that the video was being correctly digitized and it was getting to the FPGA correctly. The next task was to design the synchronization logic and implement it into FPGA. Before describing the decoding of the digital video signal, it is important to understand the analog video and digital video formats which are presented next.

4.5.1 Analog Video

There are many video signal standards in use worldwide. The analog video system in use today (for standard definition TV) in the United States was adopted in 1940. The color standard was approved by the National Television Standard Committee (NTSC) in March of 1953. On a TV screen the image is displayed by scanning (or sweeping) an electrical signal (to produce a dot) across the screen, one line at a time, as shown in Figure 13. The amplitude of that signal represents the intensity of the dot (or

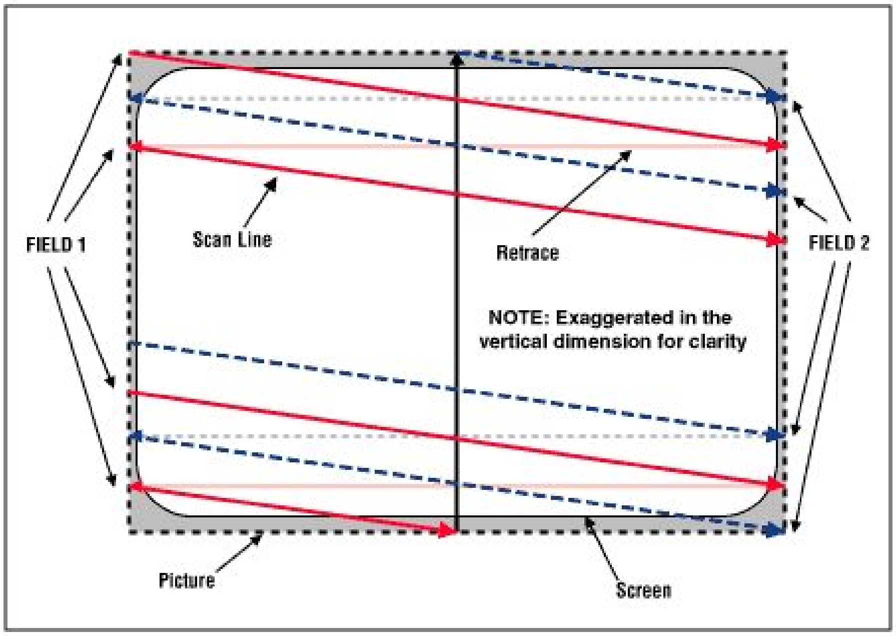


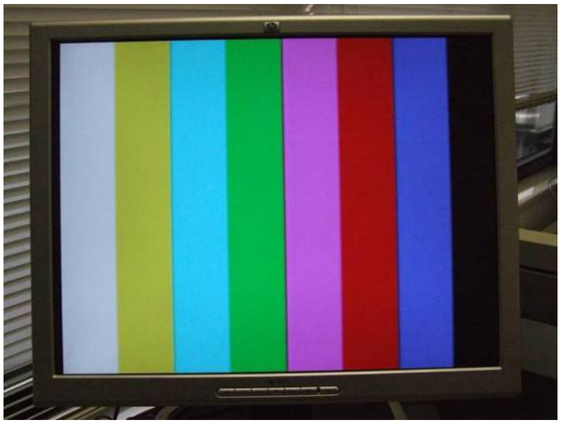
Figure 13: Interlaced Scanning System
(Source: Maxim app note video circuit)

pixel) on the screen at that instant. The display is scanned starting from the top-left corner of the screen and each line is scanned from left to right. At the end of each line, there is a portion of the waveform which is known as the horizontal blanking interval. Horizontal blanking tells the scanning circuit in the display to retrace (or flyback) to the left edge of the display. Retrace takes a very short time compared to the line scan time. While the screen is drawn from left to right, the beam (in the CRT display) is also moving downward (very slowly) which results in the next line being drawn slightly below the first line. Starting at the top, all lines are drawn that way. Once the lines reach the bottom of the screen at the end of picture, there is another portion of the waveform called the vertical sync pulse. It tells the circuit in the display to retrace to the top of the screen. The composite video signal is interlaced, meaning each picture frame is divided into two sets of even and odd lines. The set with even lines is called even field and the one with odd lines called odd field. In the NTSC system, there are 520 active lines per frame, (262.5 lines per field) and 29.97 frames per second. The NTSC standard encodes the color information using a 3.579545MHz carrier signal where the amplitude of the signal represents the saturation of the color and the phase angle (relative to a reference) represents the instantaneous color hue. Figure 14 shows a color and a black-and-white horizontal video line.

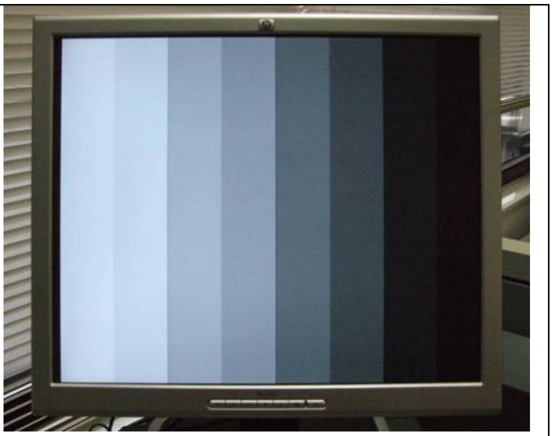
4.5.2 BT.656 Digital Video Format

The color information in the composite analog video signal is embedded in the color carrier. The display device decodes this signal to get the luminance (brightness), hue (color) and saturation (darkness or intensity of color). This information is eventually converted into three signals: red, green and blue. There are many digital video standards in use. The digital signal format most widely used in the video capture type application is BT.656 parallel interface. The BT.656 parallel interface uses 8 bits of multiplex YCbCr data (at a 27 MHZ clock rate). For each line of analog video (in NTSC format), BT.656 digital video has 1716 bytes of digital data, as shown in Figure 15 and Figure 16.

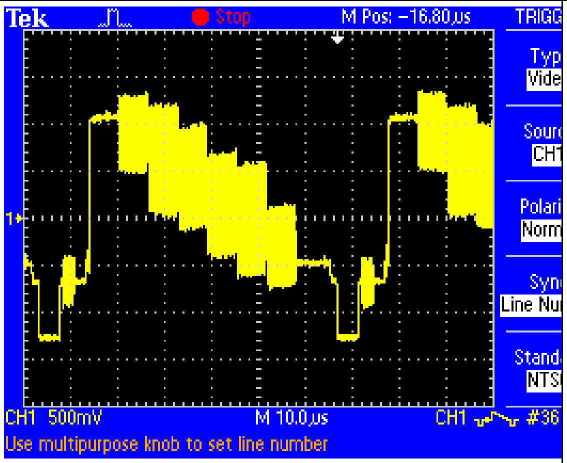
The BT.656 digital video standard specifies the active video resolution to be 720 X 486 for 525-lines/60-field NTSC signal. For each horizontal line, 1440 bytes out of 1716 bytes are for active video, and the remaining 276 bytes carries synchronization information and can carry other information such as audio, closed captioning and teletext. There are no separate synchronization signals like H-sync, V-sync and blanking, which result in a reduction of wires. Even though the number of wires is reduced by eliminating



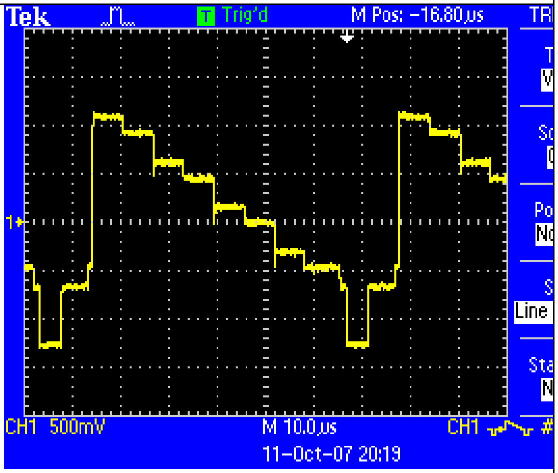
Color Bars on the Video Screen



Color Turned off.



One Horizontal Line of Video Signal for the Color Bars



One Horizontal Line of Video Signal for the Gray Scale Bars

Figure 14 : Analog Video Line

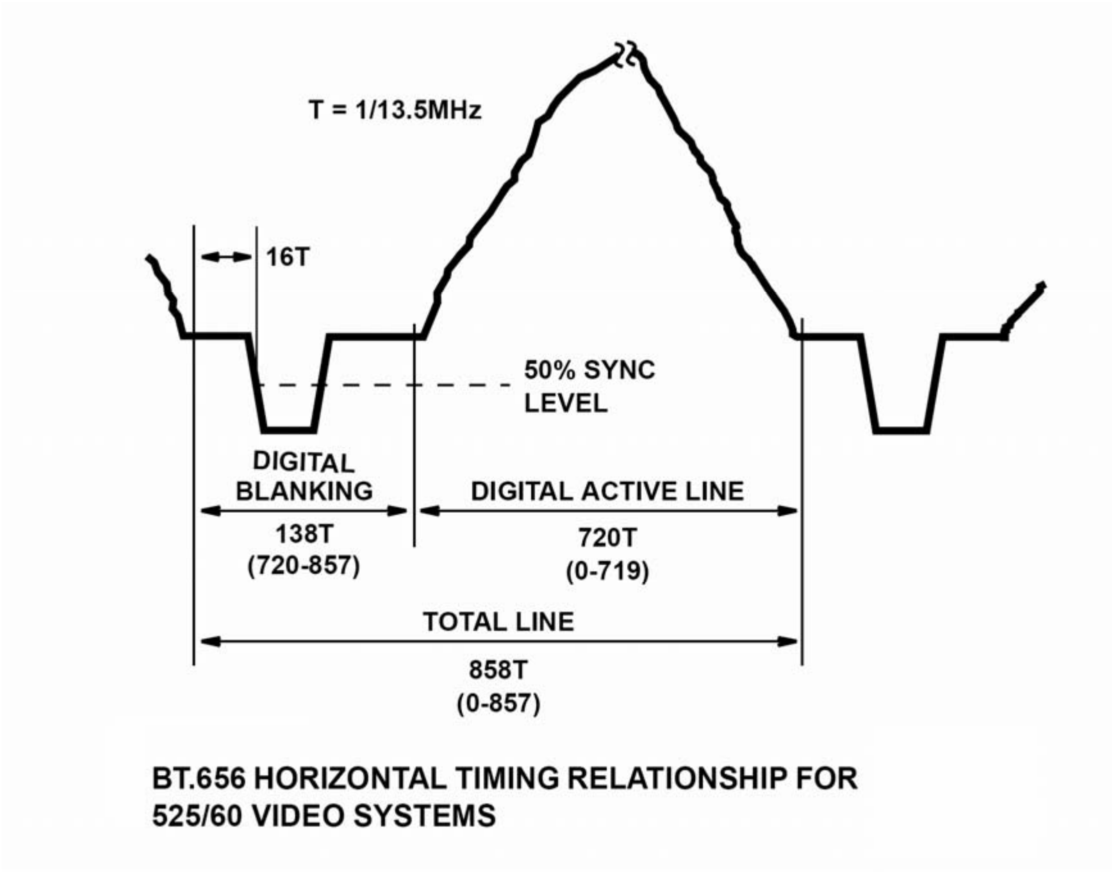


Figure 15: Analog Video Line for BT.656 Format
(Source: BT.656 Video Interface for ICs – Intersil Application note AN9728.2)

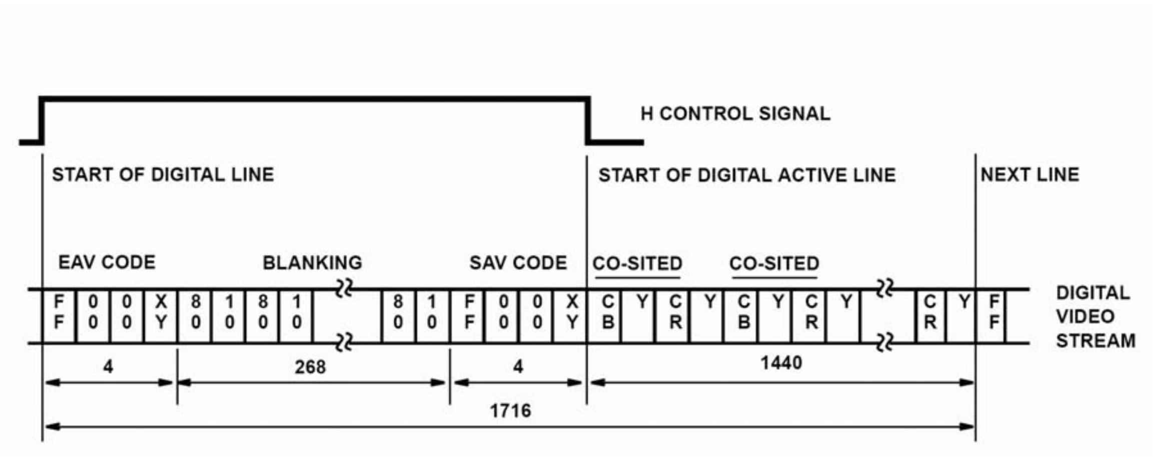


Figure 16: BT.656 8-Bit Parallel Interface Data Format
(Source: BT.656 Video Interface for ICs – Intersil Application note AN9728.2)

the sync signals, the overall complexity of the system is increased as the receiving hardware has to decode the embedded timing and sync signals from the digital video stream.

YCbCr is a color model that is used to encode the color information for digital video applications. Y is the "luminance" component and Cb and Cr are the blue and red "chrominance" components for BT.656 digital video conversion. The analog signal is sampled at 27 MHz so that there are 720 samples for each horizontal line and 2 bytes of data for each sample. The data is in the format of Cb_0 Y_0 Cr_0 Y_1 Cb_2 Y_2. The picture of a barn and mountains in Figure 17 is added to demonstrate the color image and its components into Y, Cb and Cr.

4.5.3 Video Synchronization Codes

As shown in the Figure 16 (BT.656 data format) the digital blanking pulse starts and ends with four byte EAV (end-of-active-video) and SAV (start-of-active-video) codes. These two codes (SAV and EAV) have timing information embedded in the codes and have a sequence of FF, 00, 00 and XY where the XY is the status byte or status word for a 10-bit system. The status byte $1FVHP_3P_2P_1P_0$ (where P_0 is the LSB) is defined as:

F = 0 for field 1, F = 1 for field 2 (7th bit)

V = 1 during vertical blanking (6th bit)

H = 0 at SAV, H = 1 at EAV (5th bit)

P3, P2, P1 and P0 are parity bits

P3 = V ☼ H (where ☼ represents the Exclusive OR function)

P2 = F ☼H

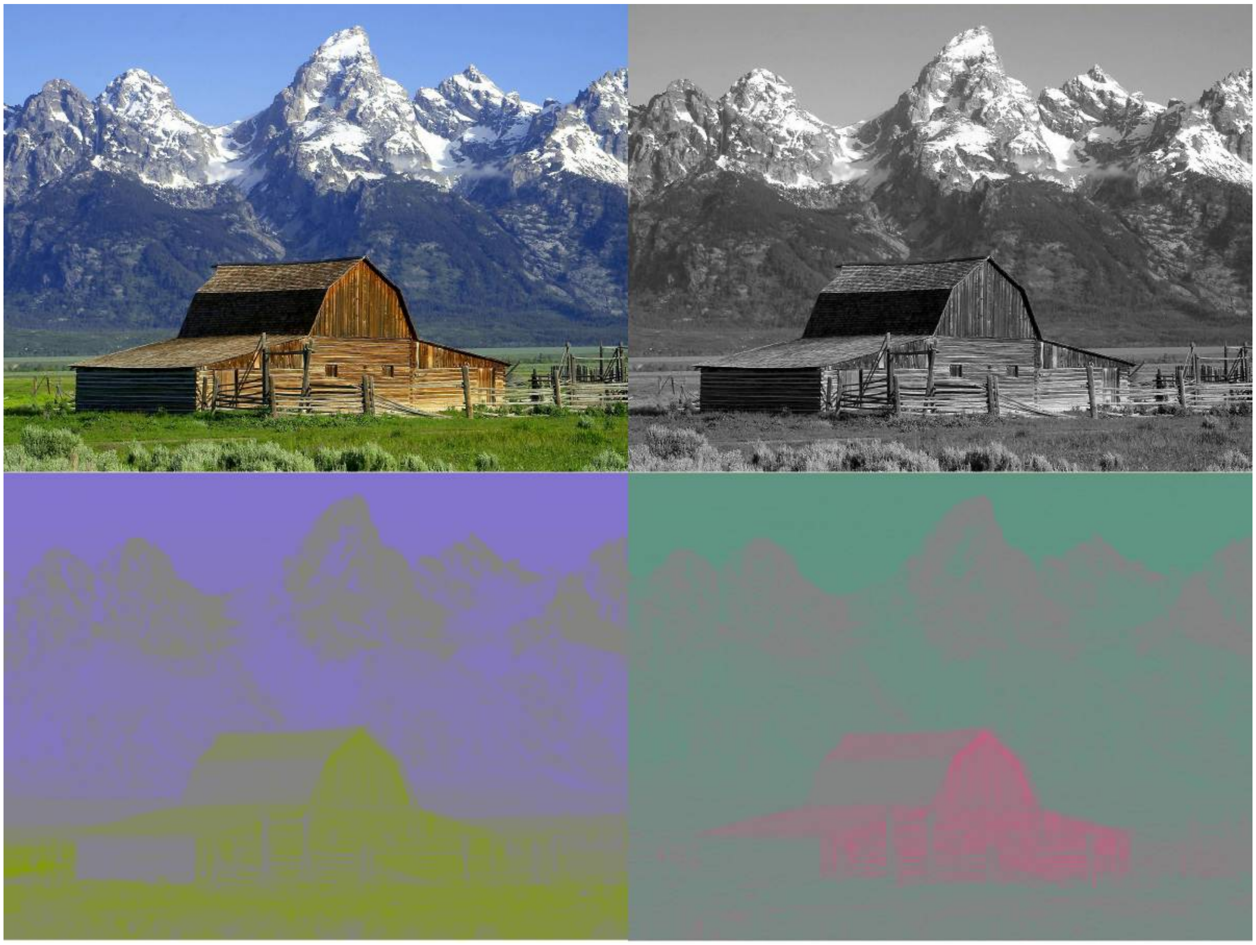


Figure 17: Showing the Color Image and its Components

Original Color Image (Top Left), Y or Luminance part (Top Right), Cr (Bottom Right) and Cb (Bottom Left). (Image taken from Wikipedia.com)

P1 = F ☼ V

P0 = F ☼ V + H

Figure 18 shows the codes mentioned above decoded for the horizontal and vertical blanking information.

4.5.4 Generating Video Synchronization Signals

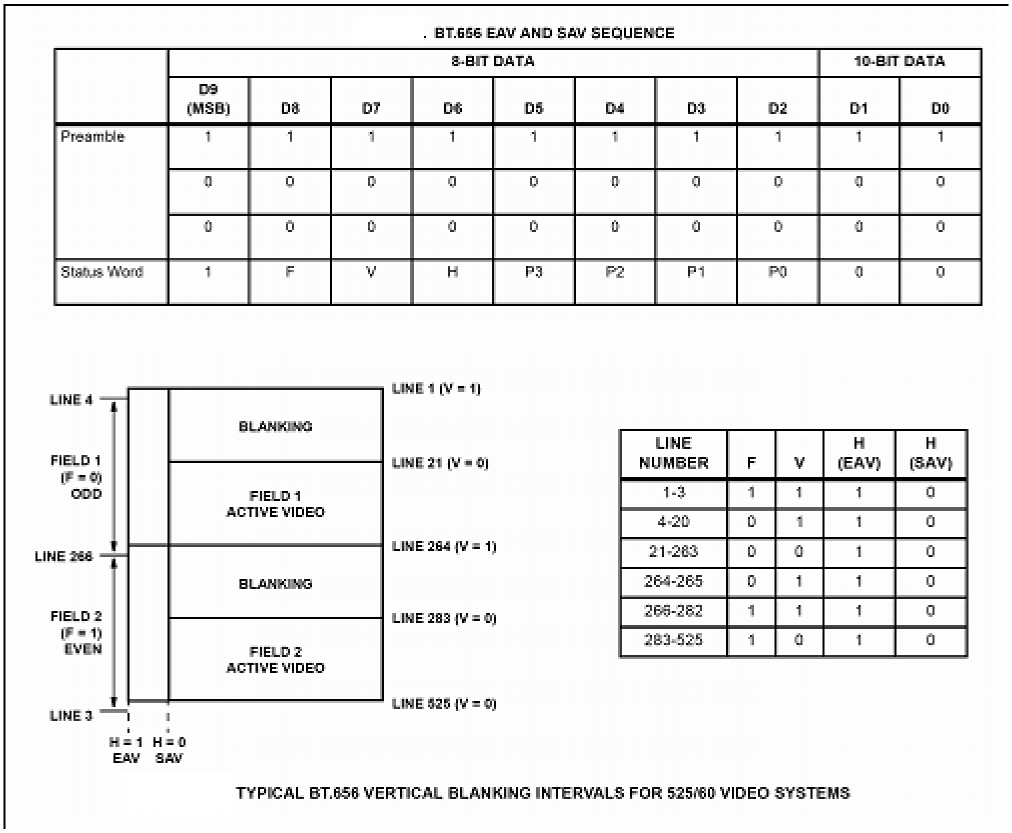
The video sync module decodes the digital data to generate the H-sync, V-sync, field and active video signals. It also assigns an address to each of the (1440 X 486) samples for each video line frame.

The following Figure 19 shows a state machine flow-chart that explains how this module recreates the sync signals that are subsequently used by the flash trigger module and the SRAM module. Once the above state machine has decoded the sync signals and identified the correct field, these signals can be used to generate and assign the correct memory address to each of the 1440 X 486 YCbCr samples.

Since the VGA monitor was used exclusively during the project for trouble shooting and verifying each module, it was decided to convert the image from interlaced to progressive scan (non-interlaced image). This was done by adding 720 to the data address at the end of each line. The even field (field 1) address starts from memory address 0 and increments by one for each clock until the end-of line. The odd field addresses start at 720 and increment like the even field address.

4.5.5 Assigning Addresses to the Data

Once the SAV and EAV codes are detected and decoded, each data point can be assigned a memory address. Since the “luminance” and “chrominance” values alternate



. BT.656 EAV AND SAV SEQUENCE

	8-BIT DATA								10-BIT DATA	
	D9 (MSB)	D8	D7	D6	D5	D4	D3	D2	D1	D0
Preamble	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
Status Word	1	F	V	H	P3	P2	P1	P0	0	0

LINE NUMBER	F	V	H (EAV)	H (SAV)
1-3	1	1	1	0
4-20	0	1	1	0
21-263	0	0	1	0
264-265	0	1	1	0
266-282	1	1	1	0
283-525	1	0	1	0

TYPICAL BT.656 VERTICAL BLANKING INTERVALS FOR 525/60 VIDEO SYSTEMS

Figure 18 : EAV and SAV Sequences

(Source: BT.656 Video Interface for ICs – Intersil Application note AN9728.2)

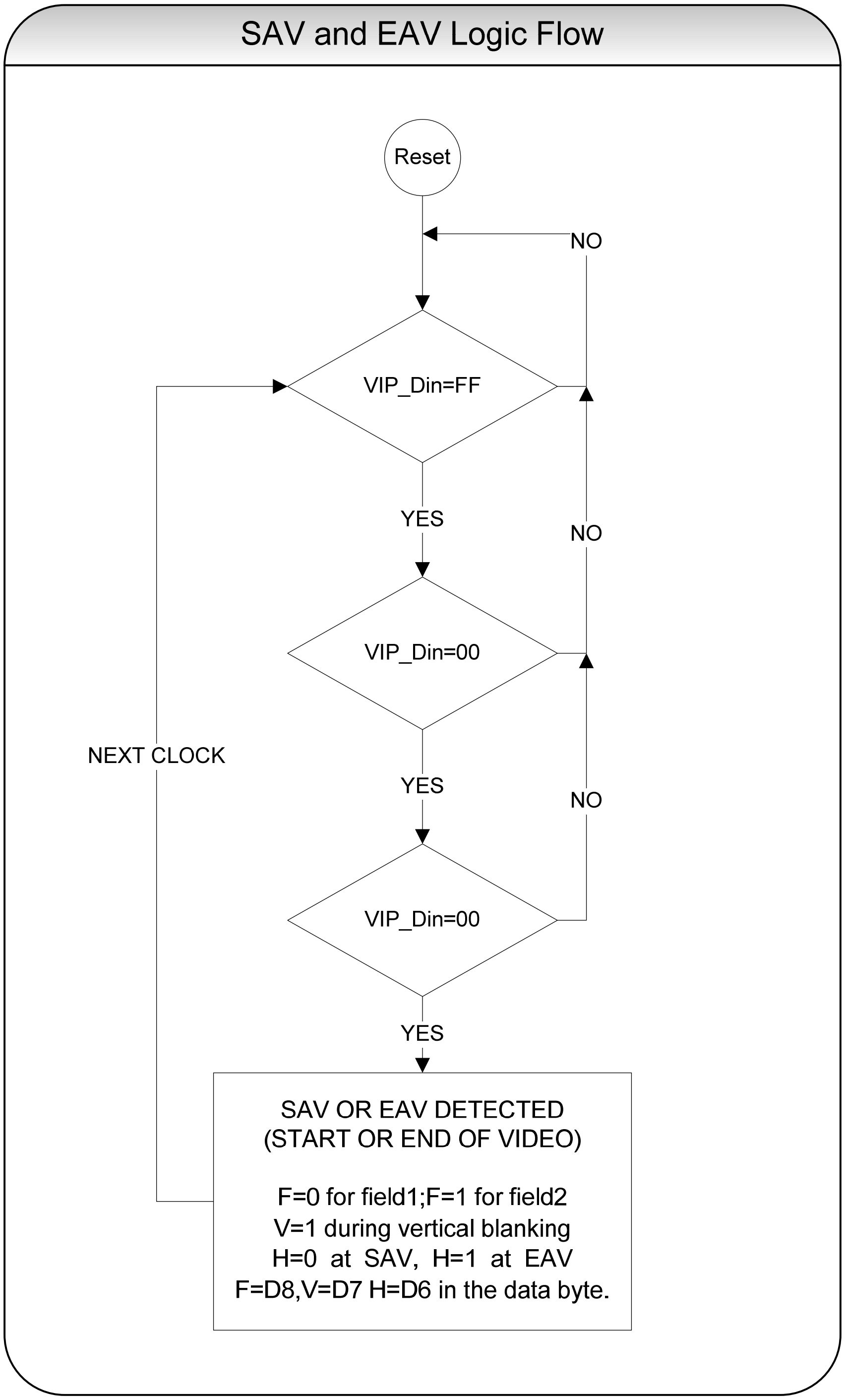


Figure 19 : SAV and EAV Logic Flow

every sample, a clever idea was adopted to separate color values from the luminance values. In the BT.656 digital video standard, for each sample the data is in the format of Cb_0 Y_0 Cr_0 Y_1 Cb_2 $Y_{2.}$--- Y_n, which indicates that the first, third, fifth and every odd byte in the active video line is a color byte, while every even byte is a luminance byte.

To separate the color and luminance, luminance bytes are written to the lower half of the memory (below 1M address space) while the color bytes are written to the upper half (above 1M address) of the memory. This way the luminance and chrominance data are separated at the beginning.

Because the analog video signal from the camera is interlaced video, the digital data is read in the alternate fields of even or odd lines. If the digital data is stored in the memory as it is received and displayed on a PC monitor or read as a BITMAP file, it will look like two half-height pictures as shown in the following figures 20 and 21. This is not an issue on the PC side where a simple C-code can manipulate the data to display the image correctly. To display the image correctly on the VGA display connected to the FPGA, the data will have to be manipulated before it can be sent to the monitor. Instead of moving data around every time it is displayed (60 frames a second), it was decided to correct this problem when the data is read from the video processor and written to the memory. The correction is made by adding an offset of one line after the end of one horizontal line. This way there is a gap of one line between two lines of active video lines in the SRAM. The lines from the next field fill this gap by adding an offset of one line at the beginning of the second frame and then repeating it (adding offset) at the end of each line. This way the two interlaced fields are stored as one progressive scan frame.

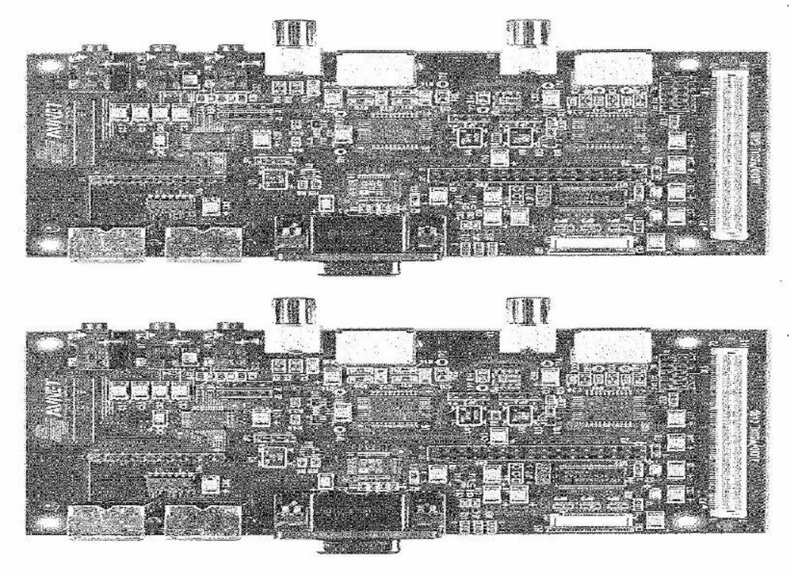


Figure 20: Even and Odd Lines of Picture

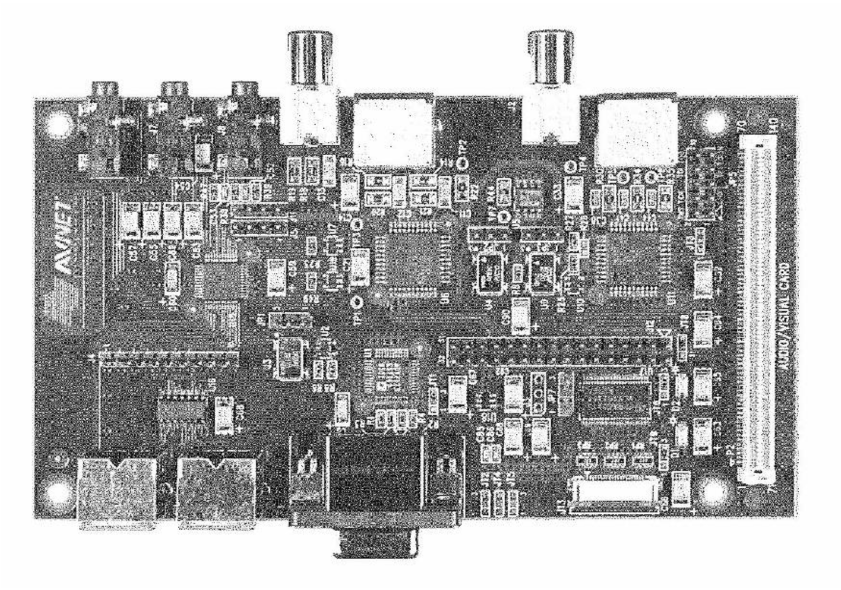
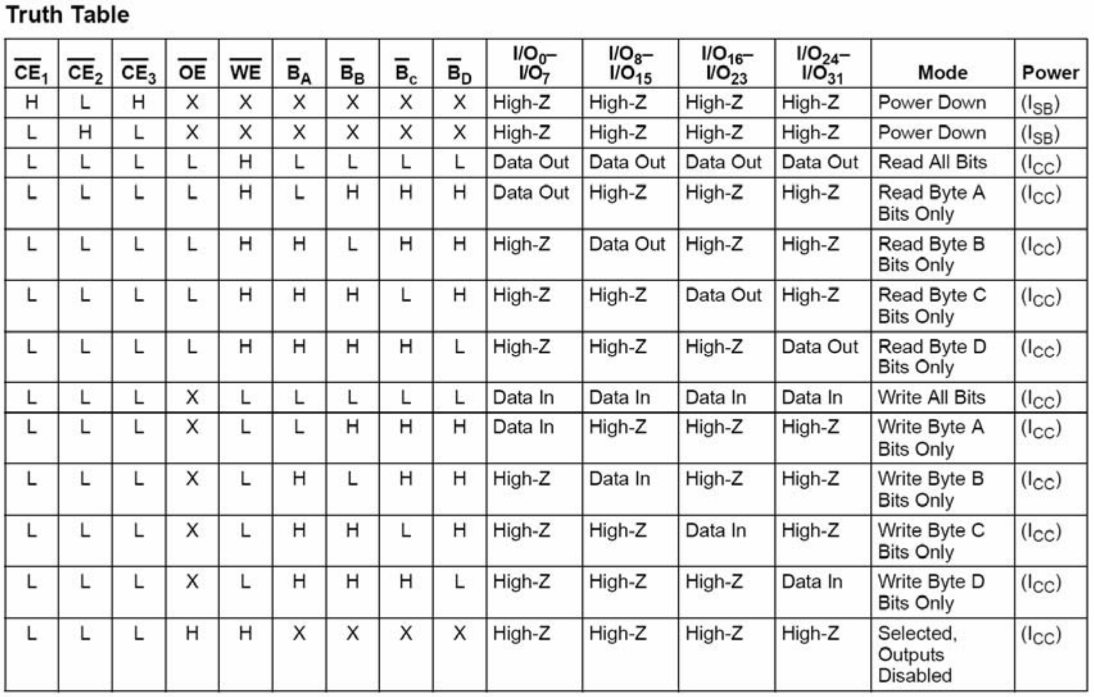


Figure 21: Complete Picture

4.6 Memory Management

4.6.1 Introduction

Until now, the memory has been treated as a linearly addressable, 8-bit wide, 2-megabyte storage space, which is not the case in reality. The development board came with 512K X 32-bit of SRAM. To store one complete frame of video, 1440 X 486, 700 kilobytes of memory are needed, which is much more in linear addressing space than the available memory address range on the board. The memory chip is organized as 32-bit wide 512K words, and each word is further divided into four 8-bit words. The Cypress chip, C47C1062AV33, also provides read and write control for each individual byte in the 32-bit word at each location. The chip provides four active low inputs (B_A, B_B, B_C and $B_{D)}$ to either read or write to any of the four bytes or disable any one or all of them. Initially a memory buffer was implemented to store four bytes (in four clock cycles) and then write it to the memory at once. The data was read from the memory in 32-bits, but only the correct byte was written using multiplexing. This scheme was soon discarded as the "video sync" module generates addresses that are not in a linear sequence (to separate luminance and color bytes). The memory logic is shown in Figure 22. The memory interface logic uses a two-to-four line decoder to use the two least significant bits of the address, A_1 A_0. To select one of the four bytes at any address, only one byte is written with the data on the data bus and the other 24 inputs remain at high impedance. To read, the same two lower addresses are used to select one out of four bytes. Thus A_1 and A_0, with the existing 20 bits of memory address, make 22 bits of addresses for the 2 Mega bytes of 8-bit wide transformed memory as in Figure 23.



Truth Table

$\overline{CE}_1$	$\overline{CE}_2$	$\overline{CE}_3$	$\overline{OE}$	$\overline{WE}$	$\overline{B}_A$	$\overline{B}_B$	$\overline{B}_c$	$\overline{B}_D$	I/O$_0$–I/O$_7$	I/O$_8$–I/O$_{15}$	I/O$_{16}$–I/O$_{23}$	I/O$_{24}$–I/O$_{31}$	Mode	Power
H	L	H	X	X	X	X	X	X	High-Z	High-Z	High-Z	High-Z	Power Down	(I_{SB})
L	H	L	X	X	X	X	X	X	High-Z	High-Z	High-Z	High-Z	Power Down	(I_{SB})
L	L	L	L	H	L	L	L	L	Data Out	Data Out	Data Out	Data Out	Read All Bits	(I_{CC})
L	L	L	L	H	L	H	H	H	Data Out	High-Z	High-Z	High-Z	Read Byte A Bits Only	(I_{CC})
L	L	L	L	H	H	L	H	H	High-Z	Data Out	High-Z	High-Z	Read Byte B Bits Only	(I_{CC})
L	L	L	L	H	H	H	L	H	High-Z	High-Z	Data Out	High-Z	Read Byte C Bits Only	(I_{CC})
L	L	L	L	H	H	H	H	L	High-Z	High-Z	High-Z	Data Out	Read Byte D Bits Only	(I_{CC})
L	L	L	X	L	L	L	L	L	Data In	Data In	Data In	Data In	Write All Bits	(I_{CC})
L	L	L	X	L	L	H	H	H	Data In	High-Z	High-Z	High-Z	Write Byte A Bits Only	(I_{CC})
L	L	L	X	L	H	L	H	H	High-Z	Data In	High-Z	High-Z	Write Byte B Bits Only	(I_{CC})
L	L	L	X	L	H	H	L	H	High-Z	High-Z	Data In	High-Z	Write Byte C Bits Only	(I_{CC})
L	L	L	X	L	H	H	H	L	High-Z	High-Z	High-Z	Data In	Write Byte D Bits Only	(I_{CC})
L	L	L	H	H	X	X	X	X	High-Z	High-Z	High-Z	High-Z	Selected, Outputs Disabled	(I_{CC})

Figure 22 : Memory Truth Table.

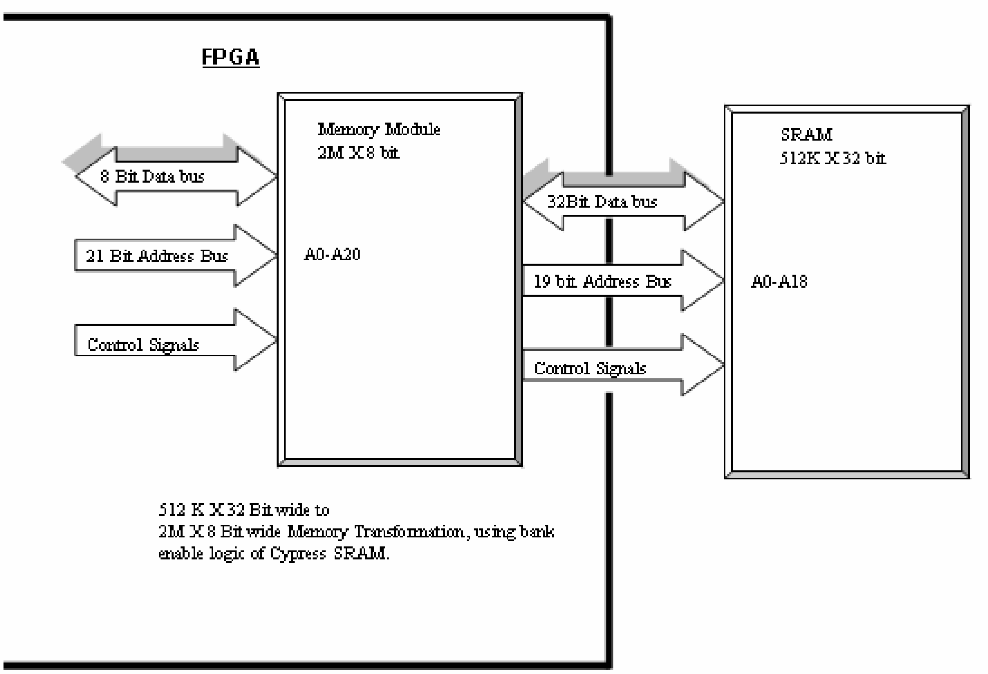


Figure 23 : Memory Interface

4.6.2 Issues with Implementations

Next, the SRAM read/write state machine was added to be able to write the pattern in the RAM and then use this data to drive the VGA monitor. Since there was no example code, it took some effort to get the SRAM code running smoothly. Initially the data from the RAM was unreadable and that was due to the lack of experience in using bidirectional I/O in VHDL. Finally, after looking at the data bus with a logic analyzer, it was realized that the RAM data bus was always latched at the last data written to it no matter what address was read. This gave the clue that the FPGA data bus was always being used as output. The fix was to float the data-bus output-buffer during the read cycle so it can be read (high Z). After being able to read and write the SRAM, it was discovered that the pattern displayed on the screen was not the same as that written to SRAM. Since there was no interface to read the data from the FPGA into the PC, it was extremely hard to debug the circuit. After filling the RAM with different patterns, it was deduced that the SRAM data bit D15 was always read 1 inside the FPGA (using a logic analyzer and an output port). Probing the circuit board, however, showed that this bit was always 0. It was concluded that there was an open circuit between the FPGA pin and the board. Since the FPGA and the SRAM are both BGA packages, it is almost impossible to verify that. When the Avnet field application engineer was consulted, he said that the FPGA was probably good as it was tested before shipping and there must be some other cause for the fault. After that, another day and a half was spent looking at every detail of the code. Finally a double entry for D15 was found in the user constraints file provided by Avnet. The first entry was correct and matched with the schematics but the second one was erroneous. Once this issue was corrected, it was able to read and write perfectly from SRAM.

4.7 Xenon Flash Lamp Triggering

Since this frame grabber (or video capture) card must work with the existing Color-Head, which is the color and trash sensor, its interface to the Color-Head must be backward-compatible. In the following section, the Color-Head or the Color-Trash Sensor will be explained.

4.7.1 Introduction

Uster Technology's color sensor is based on the original color sensor for the cotton classing which was invented by Nickerson-Hunter in 1940 who named the sensor Colorimeter. Nickerson Hunter (of Hunter's lab) also designed a Cotton Grading Chart based on the color and reflectance of the cotton. This Color Grading Chart is still the only standard for cotton color classing around the world. The color-sensor measures the color (yellowness or whiteness) of the cotton and the reflectance of the cotton sample so you can have different reflectance (or shininess) of cotton samples with the same color or vice versa.

The two quantities, color and reflectance, are measured in units of Rd and +B and completely define the color sample. Rd and +B are measured by illuminating the cotton sample at an angle of 45^o with white light. The reflected light is collected by two lenses and filtered through green and blue filters, which are then measured by a pair of photodiodes. The resulting voltages are used to calculated to Rd and +B.

Hunter's original Color-Head used incandescent lamps to measure color and reflectance. The problems with the incandescent lamp are its limited life and low optical efficiency that results is a great amount of heat produced by the lamp. Since photodiodes are very sensitive to temperature changes, these color-sensors need a very long warm-up time (1-2 hours) and have to be recalibrated every few hours (8 hours). Another problem with the incandescent lamp is its short service life during which the lamp degrades gradually so the amplifier gain has to be adjusted every week to compensate for the degradation of the lamp.

4.7.2 Xenon Lamp Based Color Measurement

Uster Technologies tried to resolve the issue with the incandescent lamp by introducing the Xenon flash lamp in their Color-Head, like the one used in still cameras. Though the Xenon flash lamp has a lot more stable output when compared to the incandescent lamp, the output is a pulse (of very short duration) compared to the flat DC-level of the incandescent lamp. To overcome the difficulty of measuring the area under the curve (energy in the pulse), Uster Technology's engineers decided to use a sample-and-hold circuit and an after-peak detector to hold the peak voltage.

4.7.3 Xenon Lamp Based Image Capture

While the use of the Xenon lamp solved a lot of issues related to the color and reflectance measurement, it made it extremely hard to measure trash by capturing a video frame through an analog camera. Since the Xenon lamp illuminates the cotton for a very brief period, the image capture circuit had to be synchronized with the Xenon flash circuit. Since ordinary video cameras have interlaced video output signals, flash timing has to be calibrated and controlled very precisely by the image capture hardware; otherwise, the two fields of video will have different illumination and the final image will have a large brightness variation between alternating scan lines as shown in the following Figure 24. To avoid this situation, the existing image capture card controls the firing the Xenon flash lamp circuit in the Color-Head. The image capture card generates the flash trigger signal and then captures the two frames exactly after a predetermined interval. The flash trigger delay is counted in number of horizontal lines starting immediately after the beginning of the vertical sync pulse at the end of the odd field. This delay is calculated by continuously triggering the flash lamp, capturing the image, and comparing the light intensity between the even and the odd fields. The time delay is varied between two extremes that are at the beginning and at the end of the vertical sync pulse. The trigger pulse position is varied using the successive approximation algorithm until the average image brightness between even and odd fields is the same or cannot be improved. The hardware on the image capture card constantly monitors the video signal in order to synchronize the flash timing with the video signal.

The video sync module generates the horizontal and vertical sync signals along with the field identifier. The following is the sequence of events leading up to the triggering of the Xenon flash lamp and the capturing of the image.

1. The FPGA receives the command to trigger the flash from the PC (through the USB controller).
2. The Flash Timing module waits for the end of the odd field.

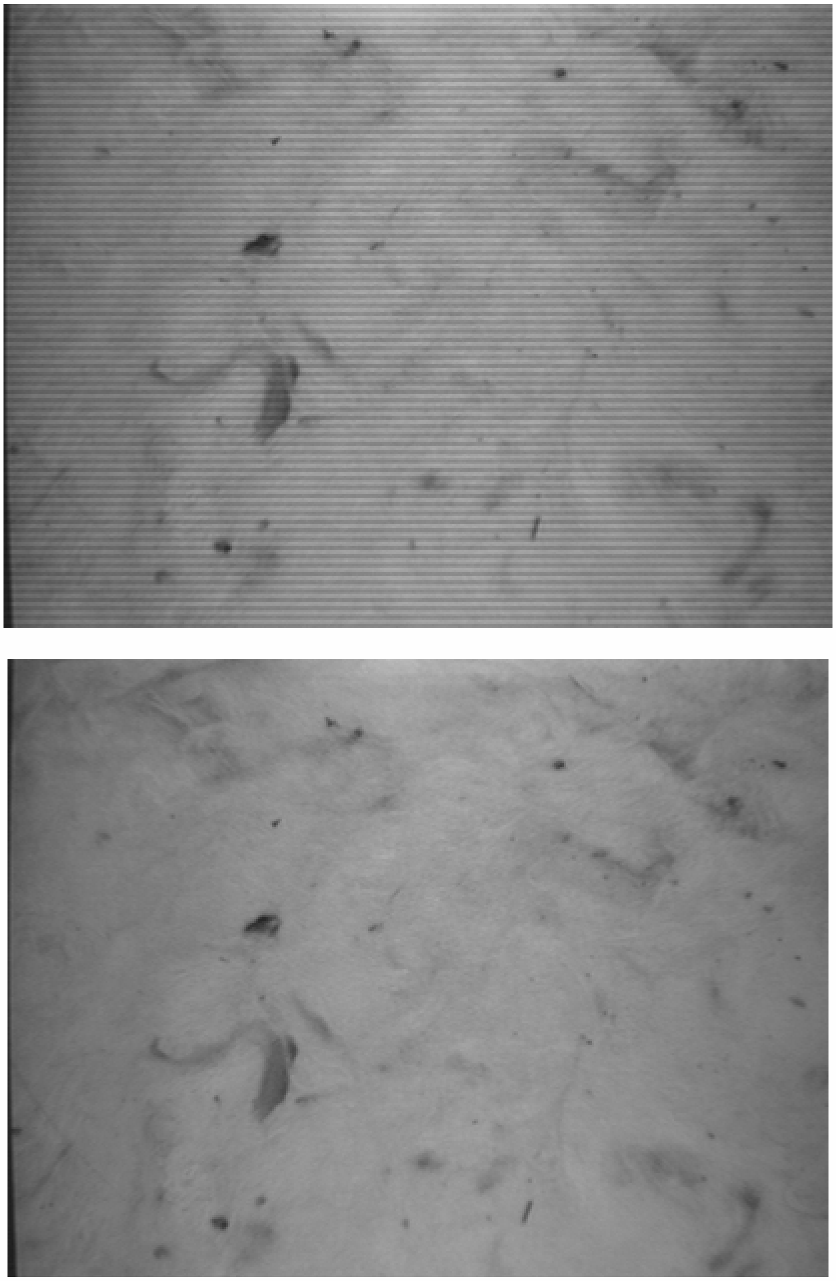


Figure 24 : Cotton Sample Image with Incorrect (Top) and Correct (Bottom) Flash Timing

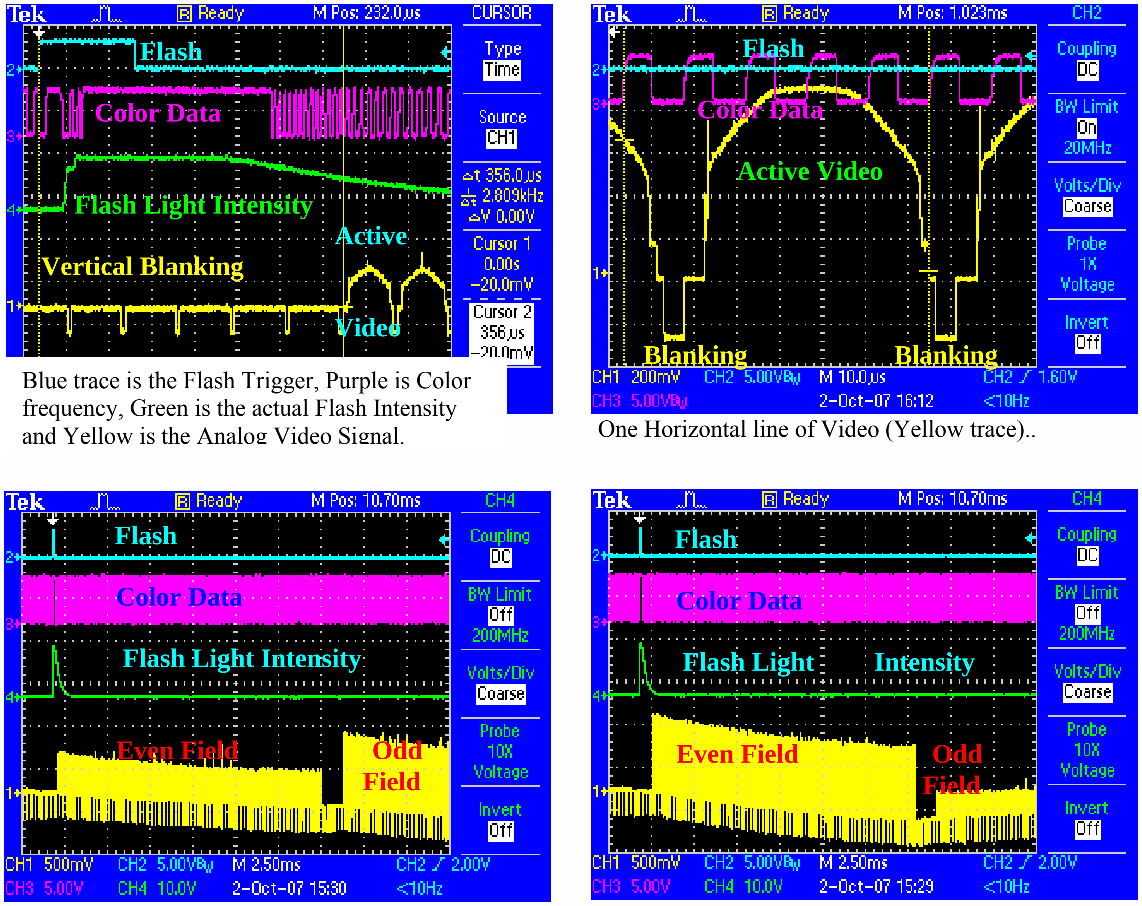
3. At the end of the odd field, the flash module waits for the vertical sync signal.
4. At the beginning of the vertical sync, the delay counter is loaded with the delay value.
5. At the end of each horizontal line, the delay counter is decremented until it is zero.
6. When the delay counter is zero, the flash trigger signal is set high for one horizontal line.
7. At the end of the next horizontal line (EAV), the trigger signal is reset and the image capture bit is set for the video capture module to capture the following two fields into the memory.

Figure 25 shows the oscilloscope screen shots for the analog video signal, flash trigger signal flash light intensity signal and the color data signal. The top-left oscilloscope screen capture shows the detail of the signal during vertical blanking. The top-right oscilloscope screen capture shows the detail of the signal during active video. The bottom-left and the bottom-right oscilloscope screen captures show the uneven field amplitude due to incorrect trigger timing.

4.8 Top Level State Machine

4.8.1 Introduction

The TOP-level state machine is the main controller, responsible for all the interaction between different modules in the design. After power up, the I2C Initialization Module



Blue trace is the Flash Trigger, Purple is Color frequency, Green is the actual Flash Intensity and Yellow is the Analog Video Signal.

One Horizontal line of Video (Yellow trace)..

Figure 25: Oscilloscope Screen Captures for the Flash Trigger and the Video Signal

starts the video input processor and video encoder initialization. The power-up reset events are initiated by an external power-on reset signal. During the I2C initialization process, the TOP state machine stays in the RESET state waiting for the Done_I2C trigger. After the VIP and Video Encoder have been initialized, the I2C module asserts the Done_I2C bit, after that the TOP module switches to "Wait_for_the _next_Command" state. In this state, the next action is dictated by the command from the PC through the USB bus, or for the manual diagnostic, by the input switches. At this state, there are multiple possibilities. Figure 26 below shows only three possibilities. In a normal image capture situation, the sequence would be to load the Flash delay and then wait for the next command. For the diagnostics the PC may request a Flash trigger, a USB read of the memory contents, or re-initialization of the video processor and video encoder.

4.8.2 Image Capture Sequence

When the command received is to capture the image, the TOP state machine waits for the start of the new video frame. Once a new frame starts, it initializes a down counter loaded with the flash delay count. For each clock tick, the counter is decremented. When the required delay has elapsed, the output bit for the Xenon flash lamp trigger is asserted causing the lamp to flash. After the flash event, the flash trigger bit is reset again and the start of new frame bit is polled. With the start of the new frame, each of the active video byte is written into the memory through the "memory interface module". Once both fields are written to the memory, the frequency counters are read and the data is written to the

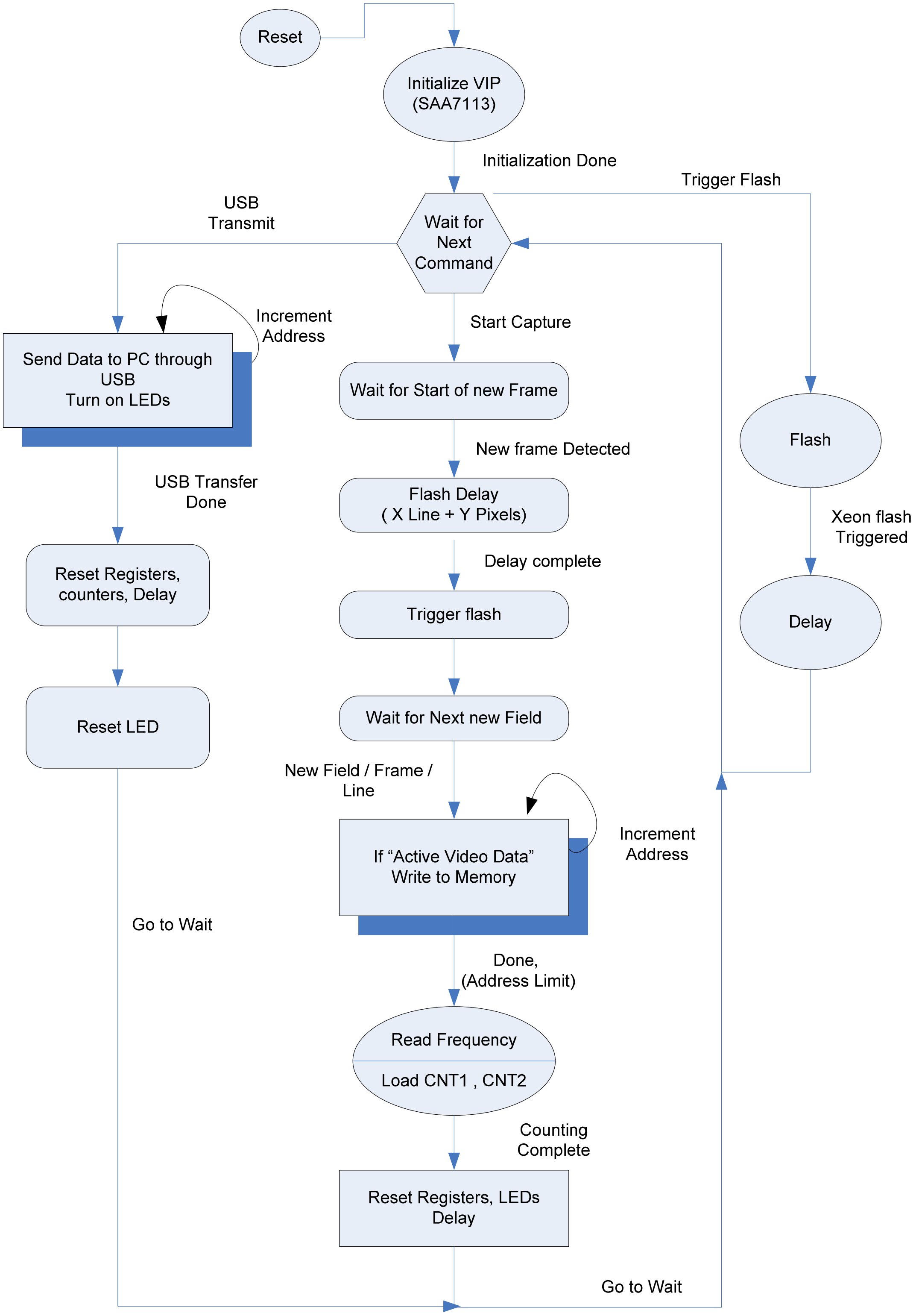


Figure 26: Top Level State Machine

memory for the transmission to the PC. If every thing goes smoothly, the status byte in the memory is written with all 1's. If there is loss of sync and the video field detection is delayed, a watchdog timer flags and forces the state machine to the next step. The status register is written with the WDT error. The WDT and the status register logic has not been implemented and tested yet. After the video data, frequency counter counts and the status registers have been written to the memory, this data is transmitted to the PC through the USB port. For diagnostics, the PC can also initiate the full read of the SRAM without the capture of the Frame. Another possibility is for the PC to fill the RAM with the data and then send it to the FPGA over the USB port. This may be useful for the diagnostic and memory test (which has not yet been implemented).

Another option for the functional test and diagnostics is to be able to flash the Xenon lamp. This command is initiated by the PC over the USB bus and is helpful in testing the Xenon Flash lamp and related power supply and trigger circuitry.

4.9 Results

The objective of this project was to develop a USB image capture and data acquisition platform that will work with the cotton color and trash sensor to replace the existing hardware. The image capture hardware platform will be used in Uster Technologies' cotton classing equipments. During the project, the following objectives were accomplished:

4.9.1 Successful Implementation of Video Input Module

One of the major tasks of this design was to thoroughly understand the workings of the analog video signal, its formats, frequencies and encoding. The legacy analog camera sends out signals that were decoded and converted into a digital picture using this video input module of the signal successfully. Major milestones in that process are listed below.

- The video input processor SAA7113 was successfully initialized using I2C sub module of the system.
- Digital video data from the VIP was successfully routed to the video decoder part of the system.
- Digital video data was successfully decoded to recreate the embedded synchronization signals.

4.9.2 Video Data Conversion and Synchronization

After the analog video data is digitized and brought into the central processing unit, the digital video data was successfully decoded into still frames using inherent synchronization signals. This "framing" was done in real time and subsequently written into memory. Major steps involved were:

- The digital video signal was given to the FPGA where the horizontal and vertical syncs, field information and SAV and EAV signals were identified in the FPGA.
- From the decoded synchronization data, each active video sample was assigned a memory address based on the X and Y coordinate position on the screen, (720 X 480 active pixels of luminance and chrominance each).

- Finally, a still frame for the captured machine is sent to memory module to be processed further and communicated to the host computer.

4.9.3 Built-In Self-Test Module

The system was required to have a self cross-checking part to be used in debugging for development purpose as well as for testing various sub-modules of the system separately. Conceptualizing, designing and implementing this BIST was completed and proved very helpful in efficient and timely completion of this system. This process also gave us great insights for future designs which will deal with video capture and processing. Major components of the BIST in the system were:

- A state machine generating VGA patterns for the standard PC monitor was designed and implemented.
- Test signals packet were generated from the BIST to be sent over to the USB module that were subsequently detected by the host computer.

4.9.4 Successful Memory Management

The Memory module was successfully implemented and tested to store the BIST-generated VGA pattern image as well as the captured and converted images from the analog camera. Major features of the module were:

- Converting 32-bit wide 512K deep memory into 8-bit wide 2M deep memory.
- A module to display memory content on the VGA monitor was designed and implemented successfully.

4.9.5 Communication Module and Host Software

PC-side software was fully implemented (by Jeff Green and Arpit Jain) but it was only partially tested. The software and USB communication channel between the Cypress USB controller and the host USB port were two other major components for the system, which are out of the scope of this discussion as they were accomplished by others.

5 CONCLUSIONS

The USB bus is fast replacing many other communication interfaces. Before the beginning of this project, the project team and the engineers at Uster Technologies had no experience in the development and implementation of the USB interface. This project opened the way to a new world of possibilities for the project team. In the project proposal, it was predicted that the techniques learned and experience gained through this project would be valuable for future projects. At that time, no one has slightest idea that this prediction will come true within months. The following section 6.1, discusses the contributions this project has made towards the success of Uster Technologies.

5.1 Contributions

The most important objective of this project was to replace the archaic ISA bus interface with a more advanced contemporary communication interface. Research at the beginning of this project showed that USB was the most suitable interface for this project. Once the design team gained experience with the USB bus through this project, it was proposed to implement the USB bus into another Uster Technologies' product. The Advance Fiber Information System (AFIS) is another Uster Technologies' instrument for cotton fiber testing. The AFIS is used extensively around the world in textile labs to test the properties of single fibers. The sensor used in the AFIS was designed in 1990 and it is mostly analog and has multiple analog outputs. The sensor interfaces with the PC through an ISA-based digital signal processing board and a PCI-based data acquisition board. The availability for the parts used in the ISA-based DSP card was worse than the

ISA-based Frame Grabber board. In December of 2006, the design team for this project proposed that the current sensor be interfaced through the USB bus to the PC. This was to be done by incorporating the DSP and DAB functionality in the sensor itself. At the time of the proposal, no one believed that it was possible to replace four relatively large PCBs by one. It was proposed that the new sensor would incorporate two or three new boards. Figure 27 shows the AFIS sensor with the main sensor electronics. In the last six months, Uster Technologies diverted resources from the USB frame grabber project to the AFIS sensor project. The new AFIS sensor design was completed successfully. Figure 28 shows the new sensor next to the old sensor. The new sensor is half the size of the old sensor and incorporates the functionality of the old sensor, the DSP and the DAB board. Figure 29 shows the ISA DSP board and the PCI DAB board.

Without the experience gained from this project, the AFIS sensor redesign would not have been attempted. The benefits to the AFIS project are:

- The big problem of procuring obsolete parts has been resolved.
- The need for an expensive ISA bus PC has been eliminated.
- Sine the hardware uses fewer parts, it is more environmentally friendly when compared to the old system.
- Cost savings are huge (but confidential), due to the elimination of two large custom-made printed circuit board assemblies.

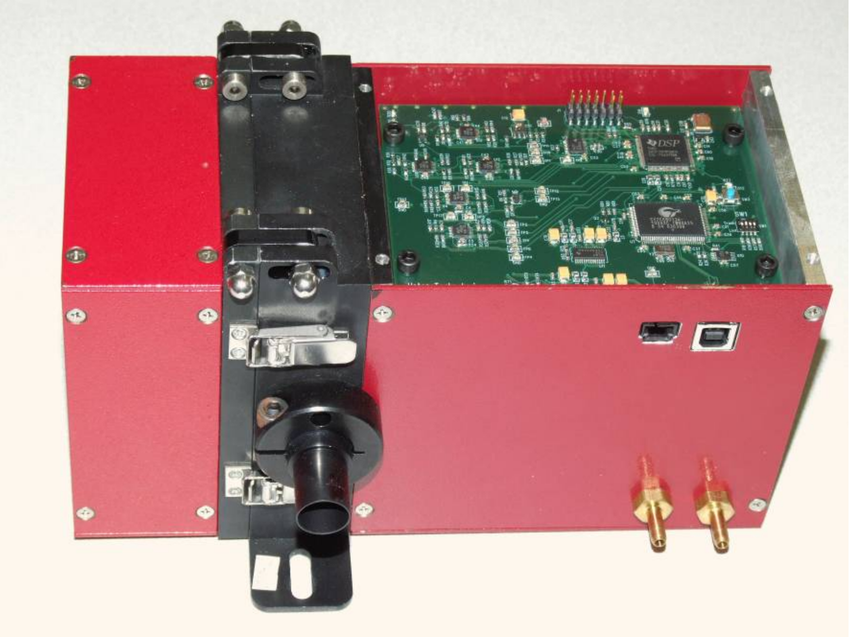


Figure 27 : AFIS Sensor with the Sensor Electronics

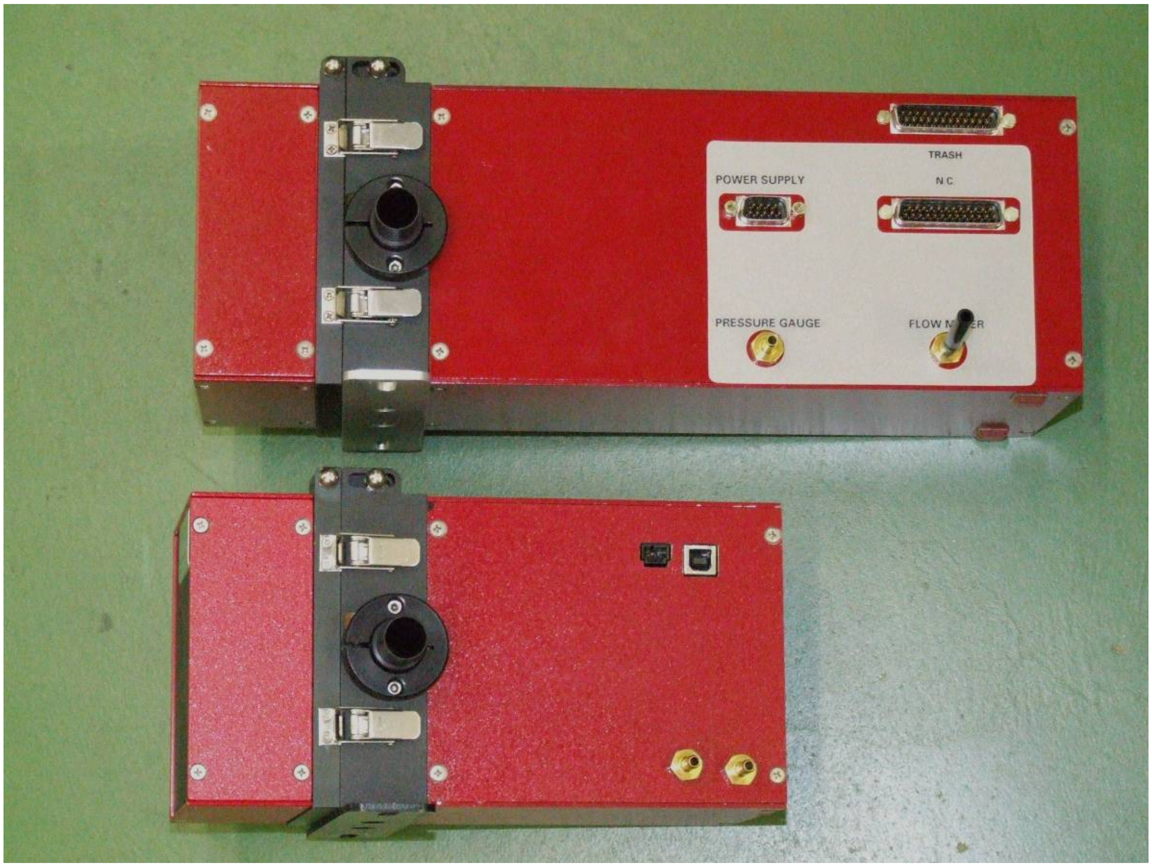


Figure 28: New Sensor Next to the Old Sensor

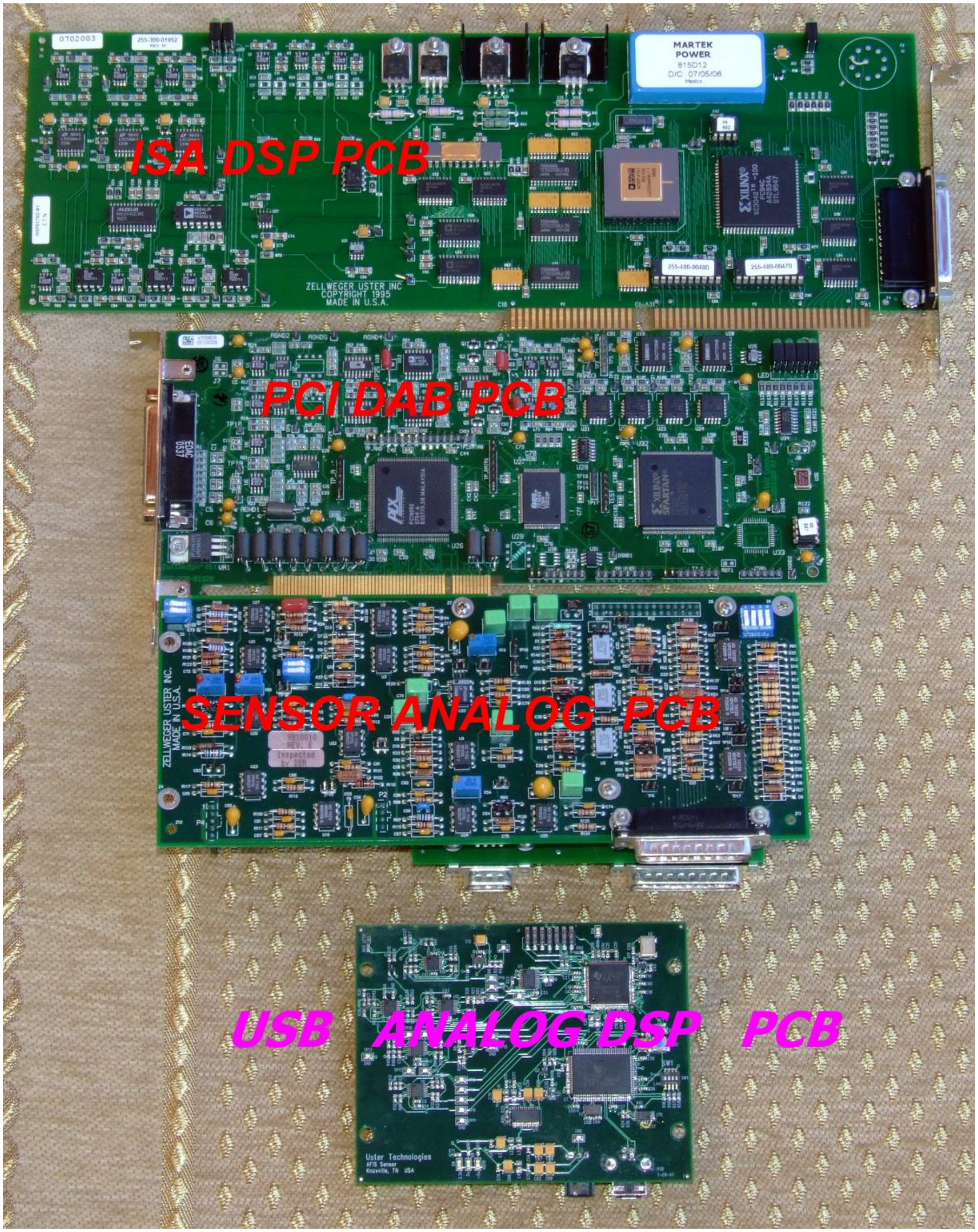


Figure 29: Three Boards (TOP) Replaced by the Single USB Analog DSP Board

5.2 Future Work and Research

During the course of the project, Uster Technologies suspended this project, initially due to the lack of resources. Recently, the management at Uster Technologies has decided to restart this project in a completely different form using a CMOS image sensor to replace the existing analog camera. Due to the lack of support and resources from the Uster Technologies, the following objectives of the project were not met:

- The image processing module was never implemented.
- The streamlined and finalized TOP module, fully interactive through the USB module, was never implemented.
- The USB module was not completely implemented on the FPGA, although the transmission part was implemented successfully but the receive part was limited to only one byte.
- The PCB design was not implemented.

Last year a project was initiated to redesign the frame grabber card. That project was reassigned a lower priority due to other high priority tasks. Recently, the camera manufacturer for the camera used on the Color-Head has announced that they will no longer manufacture analog cameras any more. There is not another supplier of analog cameras compatible with the Flash lamp. Without the analog cameras, the USB frame grabber card will be useless for future production and will have little value (for field support) due to the low failure rate of existing frame grabber cards. With that in mind, the only logical alternative is to replace the existing analog camera with a digital camera, eliminating the need for the analog frame grabber. Work has already started on the new color-trash measurement system. Figure 30 and Figure 31 show the difference the current and the future design. Figure 32 shows the block diagram of the future color and trash sensor.

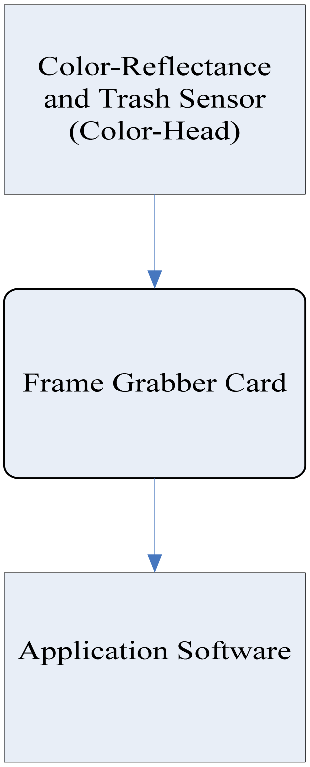


Figure 30: Major Component of Current Color and Trash Measurement System

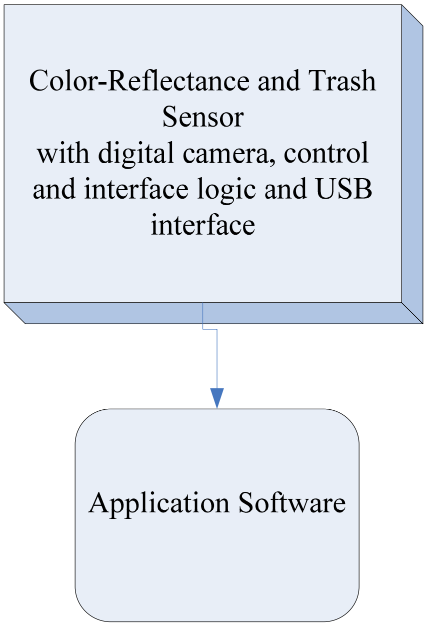

Color-Reflectance and Trash Sensor with digital camera, control and interface logic and USB interface
Application Software

Figure 31 : Major Component of Proposed Color and Trash Measurement System

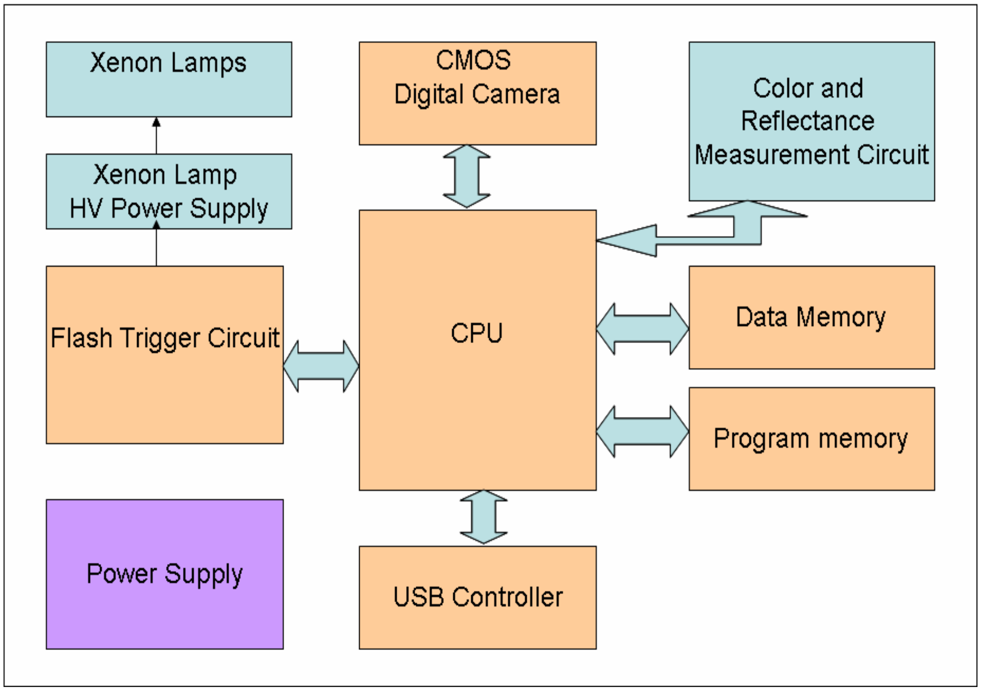


Figure 32: Block Diagram of Proposed Color and Trash Sensor

References

[1] Stefan Weidman, “Fast prototyping on personal computers,” Master’s Thesis, The University of Tennessee, Knoxville, 1998.

[2] Jorgen Peddersen, “Hardware Implementation of Video Streaming” Project Report, The University of Queensland, October 2001.

[3] Philip Jost, “Cotton Fiber Quality and the Issues in Georgia,” The University of Georgia, July 2005.
Available: http://pubs.caes.uga.edu/caespubs/pubs/PDF/B1289.pdf

[4] John Hyde, *USB Design by Example: A Practical Guide to Building I/O Devices,* John Wiley & Sons, Feb 2001.

[5] Jan Axelson, *USB Complete: Everything You Need to Develop Custom USB Peripherals,* Lakeview Research, August 2005.

[6] Keith Jack, *Video Demystified: A Handbook for the Digital Engineer, 4th edition,* Newnes; September 2004.

[7] *Audio/Video Module User Guide Rev 2.0,* Literature # ADS-003904, Avnet Corp., July 2006. Available: www.em.avnet.com/ (after registration)

[8] *Xilinx Spartan3 Development Kit User Guide Rev 1.0,* Literature # ADS-005104, Avnet Corp., April 2006. Available: www.em.avnet.com/ (after registration)

[9] *BT.656 Video Interface for ICs*, Application note AN9728.2, Intersil Corporation, July 2002. Available: http://www.intersil.com/

[10] *Soft USB Controller Design Challenges*, Application note AN1070, Cypress Semiconductor Corporation, Jan, 2002. Available: http://www.cypress.com/

[11] John A. Swanson, *Building a USB 2.0 Device AMBA Subsystem with DesignWare IP*, Synopsys Solutions Group, February 2004.

Available: http://www.synopsys.com/links/dw_usb_wp.html

[12] *The Design of a Video Capture Board Using the Spartan Series*, Application note XAPP172, Xilinx, March 1999.

Available: www.xilinx.com/support/documentation/application_notes/xapp172.pdf

[13] *Video and Image Processing Design Using FPGAs*, White Paper, Altera.

Available: www.altera.com/literature/wp/wp-video0306.pdf

[14] *"THE I^{2}C-BUS Specification,"*, NXP Semiconductors, January 2000.

Available: http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf

Appendix

USB Communication Interface

This part of the appendix is the write-up of my research partner, Arpit Jain, who worked on the USB interface and the software development. It presents the details about the interfacing medium used, the **Universal Serial Bus (USB)**. It starts with the fundamentals of USB and then describes its salient features that match the requirements of the system.

1. Introduction to Universal Serial Bus

As the name suggests, USB is a low power, high bandwidth, serial bus interface, connecting the peripheral devices using a common standardized interface socket. It is intended to replace all the existing serial and parallel ports of the legacy PCs and allow a common platform to communicate between PC to Peripheral as well as inter communication between certain peripherals. It is based on a half-duplex differential signaling method using high-speed twisted pair data cables. The USB devices are hot-pluggable, meaning they do not need system reboot to allow channel initialization as many parallel and serial channels, they get recognized on the run time and starts working with a minimum delay for recognizing the device and initializing the related controlling piece of s/w. The current version of USB (2.0 high speed) comes with a maximum speed of 480 Mb/s (60 MB/s), and is becoming the most common channel of communication for the broad range of electronic devices and PC peripherals available today.

2. Communication methodology

The Bus design is asymmetric containing a controlling hardware called host on one side and one or more (daisy chain connected) peripherals on other side. USB allows multiple devices (up to a maximum of 127) to be connected via “hubs” to one host. There will always be a root hub connected to the host and the children hubs connecting via root. The peripherals or functions have logical channels connecting to host which is called pipes. These pipes terminate on the USB devices at “endpoints”. These pipes can be in either direction, in and out of host controllers as shown in Figure 33.

The pipes are unidirectional and transfer the data to and from the endpoints. Each device can have a max of 32 pipes (16 in each direction). These pipes carry data in packets over the bus. The packets are distributed over time as “frames”. Each frame has a specific length in time and contains a specific number of bits.

The data frames are classified into three groups,

a. setup frames,
b. data frames and
c. status frames.

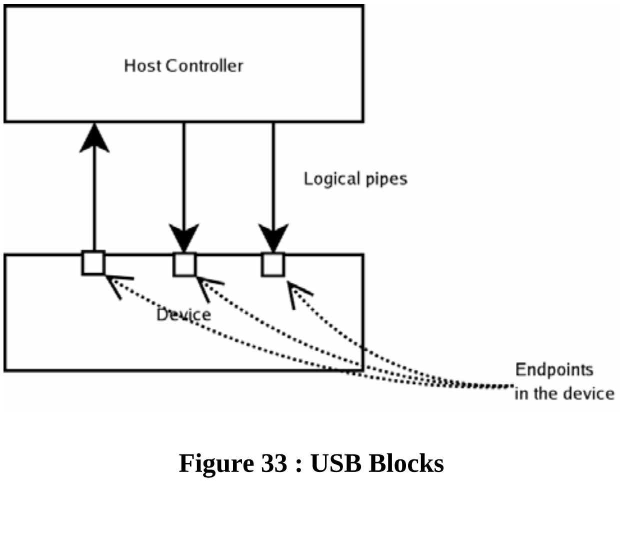


Figure 33 : USB Blocks

The names are self-explanatory and signify the function of data contained in it. Each frame starts with SOF (start of Frame) and terminates with EOF (end of Frame)

3. USB communication modes

The USB channel is able to communicate in three different modes suiting the variable need of communications. The modes are isochronous, bulk and Interrupt

- Isochronous mode: Data stream travels to and from the device (and host) in real-time, and there is no error correction. This mode is generally used for real time streaming applications such as audio and video.
- Bulk mode: In this mode, the packets of data (64-byte chunks) are transferred between the device and the host and an error checking mechanism is employed to ensure full data fidelity. This mode is used for all the general data transfer applications for example data to a printer or a mass storage device.
- Interrupt mode: in this mode, the data is only meant to transfer rarely on certain events. So the interrupt is generated and an ISR (interrupt service routine is employed to communicate the one-time immediate data transfer. This mode is generally used for human interface devices (mouse , keyboard)

4. USB Physical Layer

The USB cable contains two pairs of twisted cables carrying data and power between host and device. The 1st pair contains the power supply with one pin having 5V power and other being the ground. This allows the USB devices to be powered up by the host and no need to have their own power source. The second pair contains differential data (D+ and D-) using NRZI encoding, both carrying a signal with respect to ground, and a transition occurs when the two data lines reverse polarity with respect to one another (a Logic ‘0’ bit is represented by a signal transition and a Logic ‘1’ by no transition between the two lines). These twisted pairs are covered with a shield to make them immune to electro magnetic interference (EMI). The maximum length allowed for the USB cable is 5 Meters. The shielded cable is around 5mm in diameter. The connectors for USB come in more then 1 size. The most commonly used ones are rectangular shaped 4 pin connectors with Male and female sockets. This setup allows the high-speed serial communication with a very high noise tolerance level, which is boosted by a built-in error handling mechanism of USB.

5. Salient Features

The USB provides a communication medium for interfacing with following advantages relevant to this project.

- High speed :

The system needs to carry video information along with control signals over to the host computer. The speed offered by USB 2.0 (480Mbits/sec) was required for timely delivery of data from the device.

- Hot Pluggable :

The USB devices can be connected to the host compute with rebooting the OS. This gives a much-wanted operational flexibility to any system intended for use as an industrial device where the operation and troubleshooting of one device should not affect working of others.

- Intelligent medium:

The Universal serial bus comes with inbuilt intelligence to take care of very critical technical issues like bus termination (USB does not require termination), error handling, device assignment, and device identification and initialization. This gives it a leading edge over other available mediums.

- Compatibility :

The USB-2.0 is compatible with all new PC and does not require special industrial peripherals to be installed (like the ISA card in legacy design).

6. USB implementation:

The USB communication channel was implemented between FPGA (on board processor) and the host computer using the Cypress EZ-USB integrated microcontroller. Cypress EZ-USB controller provides a variety of communication methodologies and has one of the lowest time-to-market costs.

The USB controller from cypress –

Cypress's EZ-USB® FX2™ is USB 2.0 integrates the USB 2.0 transceiver, SIE, enhanced 8051 microcontroller, and a programmable peripheral interface in a single chip. The FX2 architecture allows data transfer rates of 480-Mbps (USB 2.0 bandwidth), and uses low-cost, industry standard, and easy to program 8051 microcontroller in a small package.

The EZ-USB was:

a. the connection between USB controller chip and FPGA
b. device identification on host side
c. Host side USB controller.

Vita

Muhammad Imran Sharafat was born in Karachi, Pakistan. He attended schools in Karachi. He attended University of Karachi for a year where he studied physics. He later transferred to the Institute of Industrial Electronics Engineering, Karachi. There he graduated with a Bachelor's Degree in Industrial Electronics in 1994, awarded by the N.E.D. University of Engineering and Technology. He worked as an electronics design engineer. He enrolled in the graduate program at The University of Tennessee, Knoxville in August of 2005 and received his Masters' degree in Electrical Engineering, in fall 2007 with a G.P.A. of 4.0.