The Super Doubler

Stephen Williams, Kenneth Richardson, Gilson Rodrigues, John Shepherd

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This paper details the motivation, goals, design, and implementation of a set-top box video conversion device targeted for classic video game consoles. The primary objective is to provide a focused solution for bridging the gap between these older devices and the requirements of modern HD display technologies. This device aims to fill the gap between cheap, high-latency solutions and expensive enthusiast alternatives.**

***Index Terms* — Analog-digital conversion, consumer electronics, field programmable gate array, image scaling, video games, video signal processing**

I. Introduction

With the move from analog video signals and CRT television sets to an all-digital world of flat panel LCD, plasma, and other technologies, many older entertainment peripherals yield inadequate results when displayed on these new sets. The reasons for this are numerous. Television manufacturers ultimately make the decision which additional connections and conversion and scaling techniques they would like to implement into the set itself. With the television industry as competitive as ever, the additional cost of considering specific devices and edge cases for a consumer base, overwhelmingly using newer digital devices, could make a large impact on units sold. Connections required for these devices are abandoned, improper scaling techniques are applied uniformly to input video sources, and several pieces of equipment can be required to merely create a connection between the console and display.

We plan to solve several of these issues with a single device. The Super Doubler exists between the videogame console and the modern television. With a 240p analog input, our device will properly scale the signal to 480p through a process known as line-doubling using an FPGA and accompanying filter and conversion circuitry. Our goal is to bridge this gap between troublesome 240p and well-handled 480p for any television set.

In addition to the video conversion functionality, the Super Doubler should be easy to use and user-configurable. The user is able to interact with the device using a remote control for basic configuration, and several firmware configuration options are to be provided.

II. Major Components

This section provides a brief summary of the components of interest selected for use in the project before the design overview in the following section. A brief description of each component's use is also provided as an introduction to the design of each subsystem in the following sections.

A. Video Decoder

In our system the Video Decoder is responsible for processing and converting the video input from the input device. The super-doubler must be compatible with a wide variety of input sources. Analog Device's ADV7181D fits the requirements of this project well. It allows for multiple different signal input formats such as composite, S-video, RGB, and the component standard YPrPb.

Additionally, the chip is extensively programmable allowing significant modification and customization of the resulting output. An in-chip color space conversion matrix can be used to convert incoming YPrPb signals to RGB color data or incoming analog RGB signals can be digitized and processed without color space conversion.. This digitized RGB data can be output over the chip's 20 pixel lines as 24-bit RGB using a special DDR mode where it is processed by the FPGA and manipulated to generate the Super Doubler's enhanced image.

B. Video Buffers

Since the target devices for the Super Doubler are all relatively old and in some cases extensively modified analog video sources, it is desirable for the incoming analog video signals to be buffered and filtered before even reaching the decoding stage. For this purpose as well as input multiplexing able to be configured separately from the decoder's in-chip mux capabilities a low pass filter video buffer was selected.

The Texas Instruments THS7353 was selected for its ease of use and configurability. With three dual input programmable channels and filter modes designed for 480i video, it satisfies our requirements. It's configuration can easily be managed over I2C and each channel can be configured independent of one another. It also features two bits of I2C slave control so that multiple buffers can be used in a single application.

C. FPGA

A field programmable gate array (FPGA) will be used as the computational core of our system. Due to real-time constraints a microprocessor would be ill-suited for the required image-processing. The inherent parallelism of the FPGA device makes it well suited for real-time processing of a digital video stream. This speed keeps latency down, a must given the sensitivity to input lag in video game scaling systems.

The specific device in the Artix-7 family under consideration is the XC7A35T. At a cost of $39.55 per unit the XC7A35T comes equipped with 33,280 logic cells, 1800kb block memory, and an additional 400kb of distributed memory. The block RAM is arranged in 36kb blocks of dual-port RAM. Dual-port RAM is useful as a frame buffer due to the ability to perform a read and write operation in the same clock cycle. There are 90 DSP48E1 slices available, each of which is essentially a multiply accumulate unit. The XC7A35T comes equipped with 5 clock management tiles which will be needed to synchronize internal logic resources to external clock signals.

D. Microcontroller

A microcontroller is used in the Super Doubler to provide system control and handle user control requests for the rest of the system. The sequential nature of control/configuration processes make them ill-suited for FPGA development. A softcore in the FPGA is an inefficient solution with such basic requirements as posed by this project for the microcontroller given the availability of many low cost adequate solutions.

The microcontroller selected is the STM32F070RB. Featuring an up to 48 MHz 32-bit ARM Cortex M0 core, 128 KB of program memory and 16 KB of SRAM for under $5, it is a powerful alternative to the traditional 8-bit Atmel and TI devices that are typically used in this space. In the T6 revision selected, there are up to 51 IO pins including two I2C hardware interfaces, four UART hardware interfaces, and two SPI hardware modules in an LQFP64 package. This provides plenty of IO access for the control interfaces needed in this project.

E. Infrared Receiver/Remote

The user is able to control the Super Doubler's basic operating features by use of an infrared remote control and infrared receiver combo. The IR receiver selected is the Vishay TSOP38238 due to its simple three pin 0.1" spacing package, 3.3V support, easy availability from many retailers, and support for common 38 kHz IR signals. A basic remote sourced from Adafruit using the NEC protocol is used in the project, but support for additional protocols would require relatively small software tweaks.

F. SD Jack/Card

An SD Card jack is provided in support of firmware updates to the system. The microcontroller interfaces to the SD card to manage these firmware updates. Due to lack of an SDIO hardware interface the SD card is interfaced to using SPI mode from the STM32F0. Since very high speed reads and writes are not required for this project, the SPI interface is more than capable of fulfilling our needs. The microcontroller also uses the SD card for storage and retrieval of device configuration settings for the user's convenience.

III. System Design Overview

A block diagram of the system level design is shown in Fig 1. The system is mostly broken into four stages consisting of the video input stage (buffers and decoder), video processing and output (FPGA), system control (the microcontroller, communications, and reprogramming interfaces) and user interface (IR remote and status LEDs).

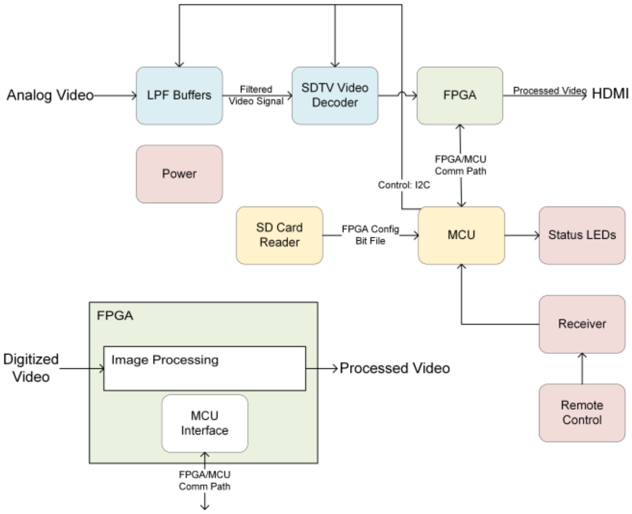

Fig. 1. System level summary block diagram showing all major components of the project and how they interact.

The complete system requires integration of the devices selected for the Super Doubler to achieve the target functionality in these subsystems.

A. Video Input and Decoding

The Super Doubler is primarily focused on supporting video connections and standards that have been left behind and were never proper standards to begin with. This led the group's motivation to support the most connections possible, but in the interest of quality we do not bother supporting low quality inputs such as single wire composite.

This leaves us with the two most commonly used connections by the market most interested in similar devices, analog component (YPbPr) and SCART, which can carry both analog RGB and analog component signals. These signals are supported by the ADV7181D in the operation required by the FPGA for the desired image scaling application.

B. Video Processing

The Super Doubler supports improvement of legacy device output by better controlled scaling of the video source than traditional (and improper) deinterlacing techniques applied to non-standard progressive signals by modern televisions. A basic line-doubling algorithm is employed as detailed in the FPGA Logic section to better convert the progressive scan source video from the consoles.

This is accomplished by the FPGA, which samples the digitized video data from the ADV7181D. In addition to the video scaling application, the Super Doubler provides for HDMI output of its improved image. This formatting and output is also accomplished internally using the FPGA.

C. System Control

The microcontroller's primary responsibility is to ensure proper configuration of the surrounding special purpose ICs and perform overall system control. This entails configuration of the discrete ICs (buffers, decoder). Initial configuration is important in that it must remember the user's previous settings to avoid inconvenience. Additionally, the system must respond to configuration requests sent by the user quickly during the device's runtime operation.

The control portion of this logic is handled via I2C in the case of the video buffers and decoder as well as GPIOs used for interfacing to the IR receiver using the STM32's timer and interrupt modules for pulse measurement.

D. User Interaction

There are two primary modes of user interaction in the system. The user controls several aspects of the Super Doubler's configuration using an infrared remote control. The Super Doubler itself produces feedback about the system status using LEDs controlled by the microcontroller. These LEDs can be used for both error code indication or to inform the user about the current operation of the system.

IV. FPGA LOGIC

The FPGA used in the Super Doubler handles all video processing once the digitized video leaves the ADV7181D decoder. It is in this stage that the video signal undergoes resolution scaling. Due to the low resolution and high contrast of the source material, nearest neighbor scaling was chosen. Bilinear or bicubic filtering would not only introduce processing latency, it would also cause image blurring which is highly undesirable. The overall video processing pipeline is shown in Fig 2. In the first stage, the video is decoded from its 12-bit DDR format into a useable 24-bit SDR format. This is accomplished using a simple state machine. All FPGA logic is written in Verilog HDL.

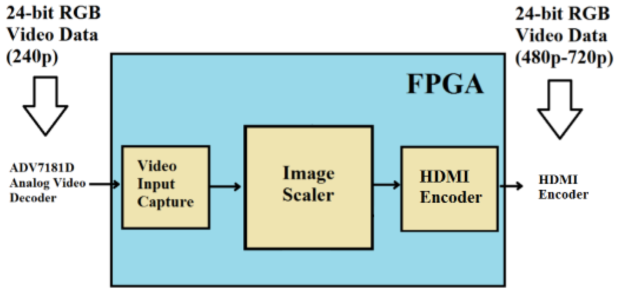
24-bit RGB Video Data (240p)
FPGA
24-bit RGB Video Data (480p-720p)
ADV7181D Analog Video Decoder
Video Input Capture
Image Scaler
HDMI Encoder
HDMI Encoder

Fig. 2. Block diagram for FPGA video pipeline.

The primary signals that the FPGA uses from the video decoder are LLC_CLOCK, horizontal synchronization, vertical synchronization, and data enable. The LLC_CLOCK is twice the rate of the pixel clock and we therefore receive two pixels every clock cycle. Horizontal and vertical synchronization are used to determine when a new line or new frame arrives. Data enable is used to gate the clock signal and prevent us from writing front/back porch information to the frame buffer.

A. Image Scaling

Nearest neighbor scaling, otherwise known as line doubling is the process of duplicating every pixel horizontally and duplicating every pixel row vertically. Line duplicating only works well with integer scaling factors. Non-integer scaling factors result in varying pixel shapes and sizes. A visual description of line doubling is shown in Fig. 3.

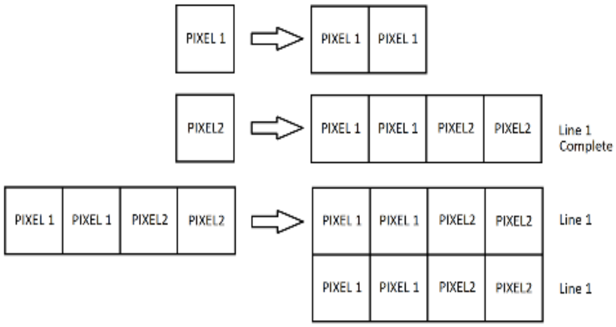

Fig. 3. Visual depiction of line doubling.

For interlaced content we will use a method under the category of field extension deinterlacing. In this method each of the two image fields in an interlaced frame are line doubled individually and displayed sequentially. This allows for the fastest processing of interlaced content while not disrupting the scaling cores operation. The downside is that a bobbing effect in which the image appears to "bob" up and down very slightly is produced. This effect is overall negligible due to the infrequent use of interlaced content in our target video game consoles.

All video content is initially written to a line doubler. In order to accomplish pixel duplication we concatenate a pixel with itself when writing to memory. For example, if Pixel_1 = (R1, G1, B1), then Pixel_1_Doubled = {R1, G1, B1, R1, G1, B1}. This allows us to do horizontal scaling in a single pixel clock cycle. When an entire image row is complete, it is then written twice to a larger buffer. It should be noted that the final large buffer is not like a traditional buffer which introduces latency, it is more alike to the RAM in a RAMDAC video output system in that it is the memory space directly read by the video transmission hardware. The overall latency of our scaling operation is roughly the time of one image row.

B. Video Synchronization and Transmission

In order to properly track the incoming video data, a Xilinx clock management block is used to lock onto the incoming pixel clock and use that to produce the rest of the video timing required in the system. This includes the output pixel clock which drives the horizontal sync, vertical sync, and the HDMI serialization clock.

To determine when a new frame has arrived, the FPGA uses the video decoder's vertical synchronization signal as a trigger. Upon system startup, this triggers the NEWFRAME flag which allows the memory buffers to be written to. After the system has begun operation, the VSYNC signal is used to reset all memory addresses. This is done to ensure that the system writes to the beginning of memory upon each new frame.

The FPGA also monitors video activity and resets operation after a certain period of time. This is done using the FPGAs internal system clock and several counters based upon if active video is detected. This is so that in the event that a input system is powered down, the FPGA will then reset the video logic to prepare for another input system.

Video is transmitted using the HDMI standard. Our target output resolution is 640x480p which requires a 25MHz pixel clock and therefore a 250MHz serialization clock for HDMI. We do not make use of any communication between the TV and the HDMI transmitter block. The reason for this is that our target resolution of 640x480p is within the HDMI standard and therefore all HDTVs should support it well.

The overall signal processing block at the HDMI output stage is shown in Fig. 4. The incoming video data from the processing and line-doubling block are captured in the HDMI output block along with the necessary signals for frame synchronization.

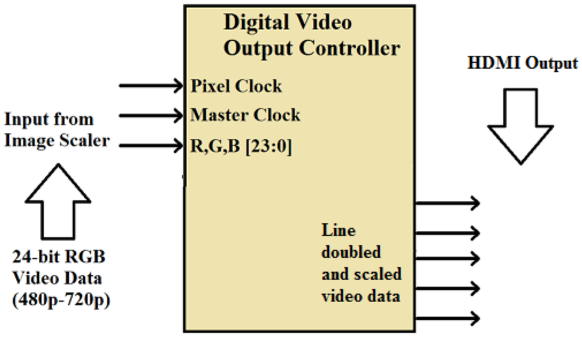

Fig 4. Signal mapping from the video scaling section to the video output logic section

V. MICROCONTROLLER SOFTWARE

The microcontroller used in the Super Doubler is responsible for virtually all system configuration and control responsibilities. This includes in-system programming for firmware updates, power on configuration of all devices, and responding to user control requests over the infrared interface. A logical connection diagram for the MCU interfaces is shown in Fig. 5.

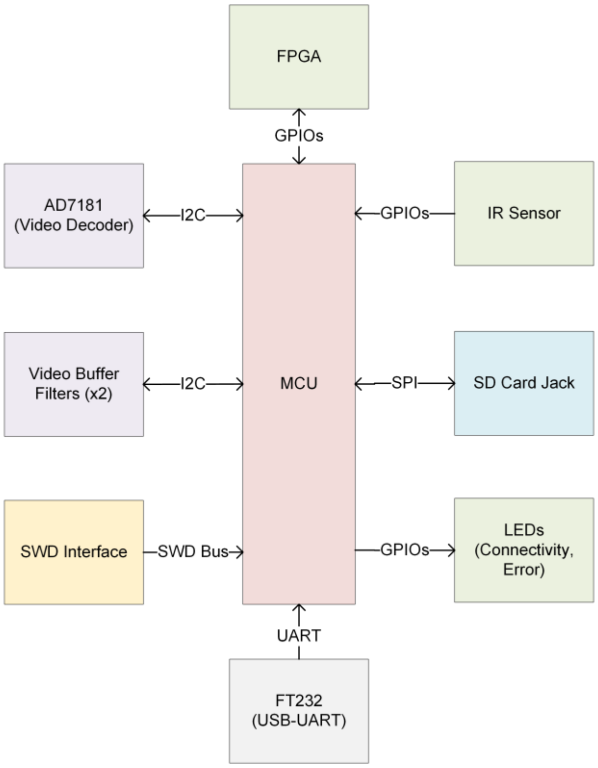

Fig. 5. Block diagram showing the logical connection view of the microcontroller, its peripherals, and the other devices in the system.

All microcontroller code used in this project is written in C. The Keil MDK ARM toolchain and development environment was used for development thanks to an expanded code size license granted for use with STM32F0 microcontrollers. An ST Nucleo development board was also used for prototyping before designing and acquiring the microcontroller PCB. The Nucleo supports SWD by an onboard ST-LINK emulator that can be accessed over USB for easy use with minimal configuration.

Several libraries were used to simplify development of peripheral functionality. The ST Standard Peripheral Library provides a register abstraction layer to the configuration and use of the STM32's hardware peripherals like GPIOs, hardware serial interfaces, and timer modules. FatFS is used to provide a FAT filesystem abstraction for use with the SD interface. Finally, ST's example libraries for STM32 evaluation boards were used as a reference for several interfaces including the infrared receiver.

A. In System Programming

The Super Doubler is designed to support full firmware updates both using developer specific tools and using more end-user friendly methods. For developer configuration and debugging of microcontroller software, the ARM Serial Wire Debug (SWD) interface using ST's ST-LINK technology is exposed via a 5-pin header configuration on the microcontroller PCB. SWD is a smaller pin count more fully featured alternative to traditional JTAG programming methods and supports both chip flashing and interactive debugging using breakpoints and watches.

For end-user configuration, use of a single jumper to pull the STM32F0's BOOT0 pin high allows UART programming of a binary or hex file generated via the development toolchain using the microcontroller's factory programmed bootloader software over the MCU's UART interface. An FTDI FT232RL USB to UART IC is used to expose the relevant pins to a micro USB connection. This interface can also support custom flashing software written to use ST's bootloader software interface. The FPGA can be reconfigured by loading a .bit file from the SD card on power on.

B. System Component Configuration

One of the two main roles for the microcontroller in this project is to provide complete system initialization on startup. This requires several steps including performing any post-programming setup for the FPGA configuration after it is loaded, initializing all special purpose ICs, and finally transitioning into the functional and user-controlled operation of the system. Fig. 6 shows a high level flowchart demonstrating the major steps of this initialization process. For convenience, the system loads the most recently used configuration from the SD card at runtime.

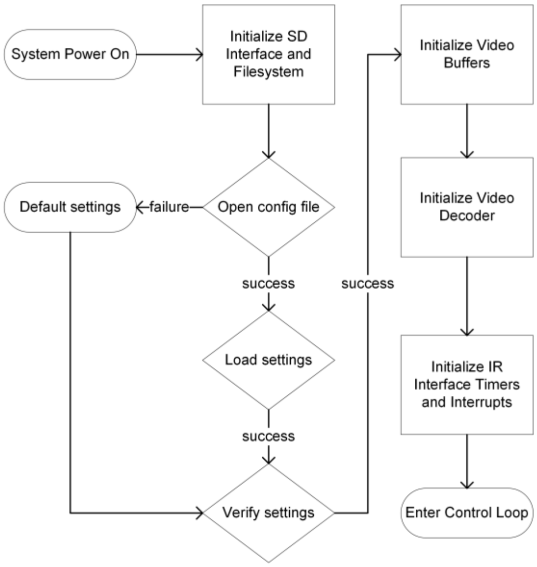

Fig. 6. Flowchart showing the sequence of major steps carried out by the microcontroller at power-on initialization.

After the initialization routine described above during power on, the microcontroller shifts focus to handling updates to this initial configuration. The user controls the configuration of the various system properties according to the details and options expanded upon in the next section. This process repeats indefinitely during device operation as shown in Fig 7.

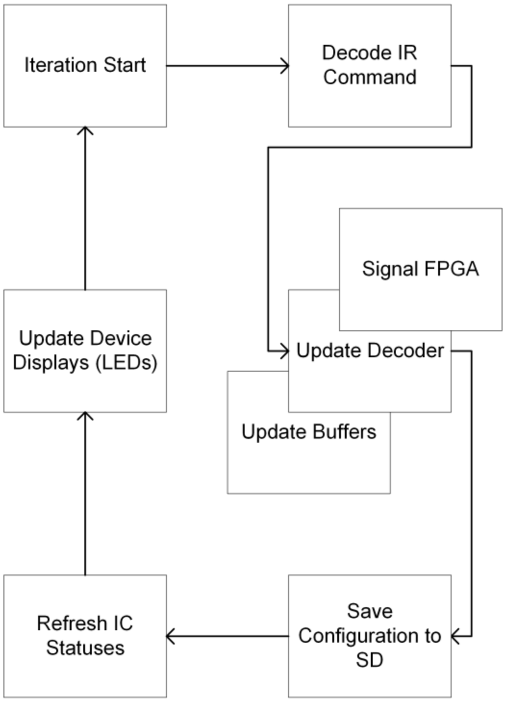

Fig. 7. Flowchart showing an outline of a single iteration of the microcontroller control and configuration loop

C. User Control

The microcontroller is also responsible for handling control and communication between the Super Doubler and the user. The user is able to configure the device using an infrared remote control (supporting 38kHz operation).

The configuration operations include basic items like input selection, some picture adjustments, and enabling/disabling some of the Super Doubler’s features like scanline emulation. Fig. 8 provides a visual summary of the basic features supported on the remote used for development. The remote control configuration options provide enough control to adequately adjust device settings for most displays and use cases.

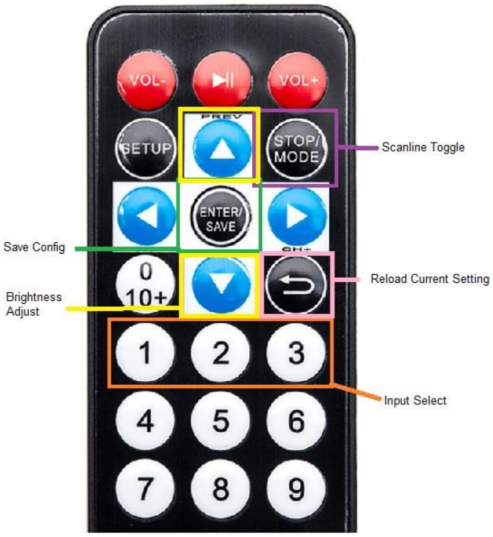

Fig. 8. The infrared remote control layout and supported settings in Super Doubler development.

The infrared signal is handled using the micrcontroller’s timer modules and received on a GPIO in input capture mode. Using this combination, the time of each signal transition from both high to low and low to high can be measured. These times can be used to then decode and verify the IR frame into the appropriate key press on the infrared remote. A valid frame can then be polled for during every iteration of the main control loop. A summary of the process as it fits into the microcontroller control logic is shown in Fig. 9.

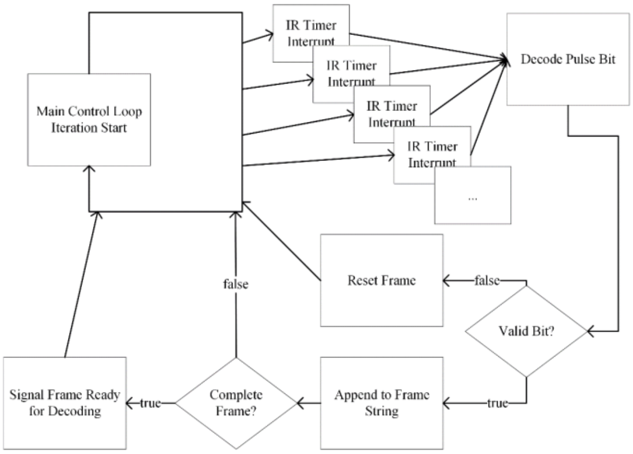

Fig. 9. Flowchart showing the general sequence of operations involved in decoding an IR frame. Timer interrupts allow the frame to continuously be decoded and polled for.

VI. PCB Design

Three printed circuit boards were designed for implementation of the Super Doubler functionality. One board was designed to provide a breakout and interface to the input stage video buffers and to mate with the decoder board. The decoder board was designed to provide a complete interface to the ADV7181D video decoder, as well as simple mating to the input video buffer board. The microcontroller board was designed to collect the microcontroller and all of its required peripherals and interfaces in a single unit.

A. Video Buffer PCB

The first PCB in the video processing pipeline is the video buffer PCB. This board contains a THS7353 video filter and is powered by a single 5V supply. The device has two three-channel inputs allowing the user to select the input via remote control. The THS7353 has a configurable low-pass filter which can be tuned for the various video sources we are using.

The purpose of this PCB is to remove out-of-band noise from the video signal before it reaches the video decoder. The input to the THS7353 PCB is connected via BNC jacks which were chosen in order to provide a secure, shielded connection for our video signals. The output also uses BNC connectors for similar reasons.

Due to the SCART video signal containing four channels and YPrPb requiring three, a single THS7353 is inadequate and we must use two ICs in order to filter both video inputs. This PCB was designed using a simple two-layer structure and was ordered through Osh Park.

B. Video Decoder PCB

The second PCB contains an ADV7181D NTSC video decoder and represents our most complex student-designed PCB in the system. It provides us the capability to decode and digitize a basic analog video signal. The digital video data is broken out to header pins and the pixel clock is carried over a shielded SMA-cable for reduced noise. RCA barrel jacks are used for the YPrPb video input and BNC connectors are used for the RGB/SCART video input. BNC was chosen for SCART to provide higher compatibility for the various SCART cable pinouts found in Japan and Europe. A simple passive external adapter converts SCART wiring to a four-wire BNC.

This PCB was initially designed using a four-layer structure and blind/buried vias. However, due to cost we had to switch over to a two-layer design and basic vias. All resistors and capacitors use 0805 size package for ease of manufacturability. All signal traces are 8mil and power traces are 40mil.

A split digital-analog ground plane is used with a single high-impedance connection point. There are four individual power planes which provide the various voltages requires by the digital and analog cores. The video decoder PCB was designed in Eagle and ordered through Osh Park.

C. Microcontroller and Peripherals PCB

A third PCB was designed to encompass the microcontroller-centric functionality of the system. The STM32F070RBT6 microcontroller, the FT232RL USB-UART converter, the microcontroller's SD card interface, and the infrared receiver for remote control interfacing are all integrated as part of this board. Additionally, pins are broken out to support using the microcontroller's serial interfaces, most of its GPIOs, and to support the SWD interface. The board is powered by any sufficient 5V wall jack with its onboard voltage regulator.

VII. Challenges

While prototyping of individual features progressed smoothly with the FPGA and microcontroller thanks to the use of the development boards and known good digital RGB sources, the group experienced significant difficulties in obtaining a working member-designed PCB for the video decoder. The timing requirements for the FPGA sampling and sync portion of the video processing are fairly tight. Revisions of the board designed by the group demonstrated poor performance of the output pixel clock needed for this sampling, The clock was several megahertz off from expected values and proved to be unusable, resulting in wildly out of sync sampling by the FPGA.

Additionally, support for the ADV7181D itself was not easy to obtain. Some configuration options discovered during testing through Analog Devices setup scripts intended for use with their evaluation boards resulted in nearly working video. But several times these settings were found to be completely undocumented in the device datasheet and user manual. It appears Analog Devices support often only provide support based on the same documentation available to the end user, compounding the issue and resulting in quite a lot of time wasted in trial and error testing with poor results.

These issues with the ADV7181D board design created significant setbacks in the overall system prototyping. The video decoder is arguably the central pillar of the rest of the system. Without its proper operation, the Super Doubler cannot achieve its goals.

VIII. Conclusion

The Super Doubler was a challenging project that proved to be a valuable learning experience for all group members. With little experience with signals processing and the challenges posed, the group was forced to learn quickly and apply the cumulative knowledge gained throughout our undergraduate education.

The project managed to combine nearly every aspect of electronics design, including working with microcontrollers, FPGA development, serial communications, signal processing, and extensive PCB design. The problems solved, and likewise the improvements that can be made to the current system, kept the group motivated throughout the project and to continue making improvements to the Super Doubler platform beyond the two semester timeline.

Acknowledgement

The authors would like to thank Dr. Samuel Richie for his advising and guidance on this project. We would also like to thank the UCF CECS faculty and our fellow students for the knowledge gained and experience we've had throughout our entire time at UCF.

Biography

Stephen Williams is graduating with a BS in Electircal Engineering. He has been an undergraduate member of the CAL research group for two years and has interests in digital signal processing, circuit design, and wireless technology. He plans to continue his studies in a PhD program upon graduation.

Kenneth Richardson is graduating with a BS in Computer Engineering. He has spent the past two years working extensively at the Robotics Club at UCF on IGVC and RoboSub vehicles and interned at Lockheed Martin. After graduation he plans to find work as a robotics software developer.

GILSON RODRIGUES is graduating with a BS in Electiral Engineering. He has been working as an intern with STS aviation. He recently accepted a fulltime position at Embraer in which he will be designing electrical systems for business airplanes.

JOHN SHEPHERD is graduating with a BS in Electrical Engineering. He has worked on several home projects exercising the areas of signal processing and communications. Having coursework in communications and power systems, he plans to work in the Power and Utilities industry.