The Super Doubler

Device for Improved Classic Videogame Console Output

Senior Design I Documentation
EEL4914 – Dr. Samuel Richie and Dr. Lei Wei
December 10th, 2015

Group 31
Stephen Williams – BSEE
Kenneth Richardson – BSCpE
John Shepherd – BSEE
Gilson Rodrigues – BSEE

1. Executive Summary

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 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.

For many manufacturers, the solution is to provide the bare minimum in analog device support and connector compatibility. Often the only analog connector provided is composite. Even when the connectors themselves are present, the methods used in processing and displaying them are typically applied in a broad and insufficient way. The result is an image which lacks clarity as seen in Figure 1.

Figure 1: 240p signal. TV default processing mode (left), XRGB-mini device output (right).

This project is focused specifically on providing an intermediary device between classic 8-bit/16-bit videogame consoles and modern television sets similar to Micomsoft's XRGB-mini, but at a lower cost and with a more focused feature set. These videogame systems typically output a 240p (320x240) signal over one of the aforementioned analog connections. Modern televisions often handle such 240p signals poorly, interpreting the signals as 480i (640x480 interlaced) and inappropriately applying de-interlacing techniques to a progressive scan signal.

2. Project Definition

We plan to solve several of these issues with a single device. Our device 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. Line-doubling scales the resolution by a factor of two while requiring very little processing time. While modern televisions often have trouble with 240p signals, the majority today handle 480p signals well enough for the average user. Our goal is to bridge this gap between troublesome 240p and well-handled 480p for any television set. If desired, the console stereo audio output may also be provided as input and carried over HDMI to the display for a single, convenient connection point.

Figure 2: 240p signal displayed on a CRT television with darkened scanlines (left), emulated scanlines via XRGB-mini.

In addition to scaling and converting the image, our device provides darkened scanline emulation. A darkened scanline, in the context of 240p displayed on a CRT television, refers to the darkening of every odd line due to only one of the image fields being transmitted as shown in Figure 2. These darkened scanlines are absent when a 240p signal is displayed on a digital television. This device will allow the user to darken every other picture line, providing a classic visual effect. An on-screen overlay will allow the user to simply and quickly check the device status and operating mode for making desired adjustments. This overlay will be interfaced with via remote control, allowing the user to adjust various image properties in addition to scanline control.

On the strict hardware interface side, our device will support a variety of analog video inputs. It will support component-YPbPr video, 15-pin VGA-RGBHV, and SCART-RGBS. For output, the device will support digital output via HDMI. This collection of

input connections covers nearly all common connections seen on the videogame consoles targeted for use with this device.

2.1 Project Motivation

We decided to design a video scaler because of our dissatisfaction with modern television 240p video support. Existing devices are either too expensive, or have poor performance. There exists a gap of suitable video processing products in the $150-$250 price range; we decided to seize the opportunity and create such a device. FPGAs are excellent for image processing and provide the designer with fine-grain control of computation resources. This allows us to create a custom video processing data path which performs extremely well with 240p video content.

2.2 Objectives and Goals

Our goal is to design a video processing device geared towards 240p video content. We plan to achieve this goal while also producing a device that is cost competitive in the current market. It must provide fine control over settings which impact image quality such as scaling factor and color levels. In order to preserve part of the experience of playing on a CRT television our device will have the ability to add in darkened scanlines to the image, rounded frame corners, and some non-linear lighting effects to mimic the characteristics of CRT display technology.

Image quality is largely determined by resolution and color balance. It is therefore important that our device has the ability to adjust such settings in order to produce a picture which is pleasing to the eye. The resolution will be controlled by an integer scaling factor, if the input resolution is 240p (320x240) with a scaling factor of two, then the scaled resolution is 480p (640x480). The user will be able to adjust the individual color channels in the RGB signals to a level which they personally find appealing.

While modern digital televisions are desired for their high resolution and crisp image, there is a certain nostalgia associated with the quirks of CRT display technology. When displaying 240p content, CRT televisions would have darken every other scanline; the darkened scanline effect is referred to as “scanlines” in the video game community. These scanlines and other mimicking techniques all work together to give the user an experience which merges together the benefits of modern and vintage television sets.

In order for the Super Doubler to be competitive it must provide flexible high performance video processing at a reasonable cost with an easy to use interface.

2.3 Requirements Specification

Requirements were developed at the outset of the project and refined over the course of preparing this design document along the lines of 6 key areas: physical characteristics, power consumption, device input/output connector and format support, technical

characteristics, cost, and user experience. The requirements are summarized in the list below.

Table 1: Summary of requirements specification for the project.

Physical	Weight < 2 lbs
	Volume < 25 in^3
	Complete standalone device in single enclosure
Power	Total device power usage < 20 watts
I/O Support	Support component-YPbPr, VGA-RGBHV, and SCART-RGBS
	Support LR stereo audio input
	Support HDMI output
	Support VGA output
Technical	Processing time < 30 ms
	Full 4:4:4 RGB Processing
	Resolution Scale Factor >= 2
	240p ⇔480i switch time < 2 frames' time
Cost	Device bill of materials < $150
User Experience	Support overlay for inspecting device statu
	Support for scanline emulation
	Support for configuration using remote control
	Support for FPGA reconfiguration via SD Card

These requirements are developed to facilitate specific design strategies and will be used to judge prototypes and proposed designs. Adherence to these requirements should remain absolutely critical to every decision made in the design process.

3. Relevant Research

3.1 Similar Past Products

For as long as digital television sets have been on the market, there has been a need to support devices which output analog video signals. Video formats such as composite, s-video, component, and SCART are used by classic video game consoles. While it is possible to find current TV models which support some of these format, many of them do not properly support 240p video signals. There are many existing devices which aim to improve 240p analog video support with digital TV sets. The capabilities of these devices vary greatly with some only providing very simple color space conversion, such devices are called transcoders. More capable devices provide resolution scaling, such devices are called line doublers when performing 2x scaling and full scalers otherwise. We will limit our discussion to products which are intended to be used with video game consoles as other devices produce poor results. The hierarchy shown in Figure 3 will be used to classify previous devices based on a minimum feature set. From least capable to most capable it is: transcoders, line doublers, and full scalers.

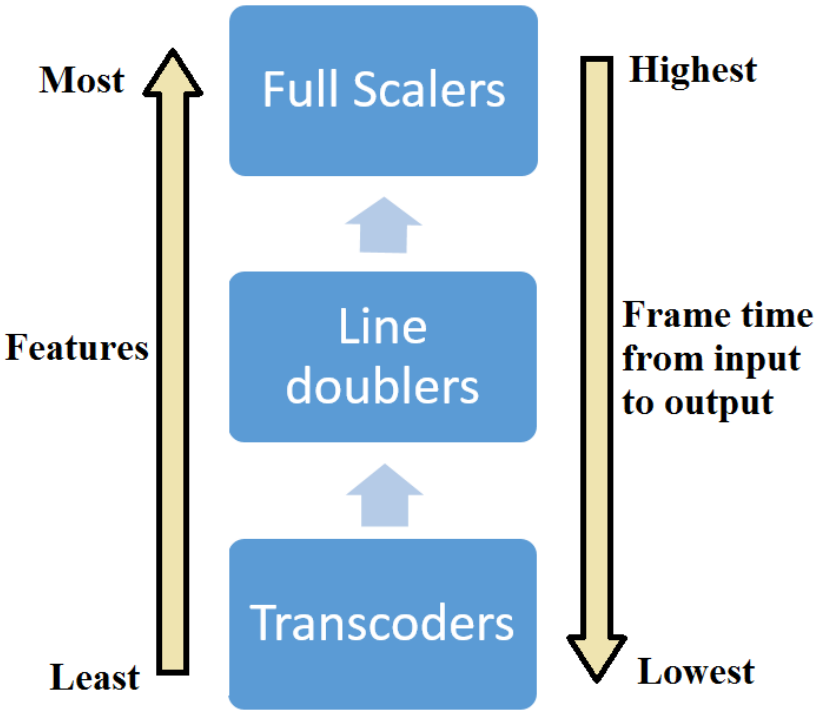


Figure 3: A line doubler represents a balance between processing time a features.

3.1.1 Transcoders

Older 240p capable video-game consoles have the ability to use RGB video. RGB video is typically transmitted using a SCART cable and is considered the best option for 240p consoles. Unfortunately the SCART video standard did not catch on in North America, with many US companies opting to support component video or s-video instead. RGB to YUV transcoders represent a low-cost solution to support your older video game consoles with a digital TV set. These devices perform a simple color-space conversion from the RGB color-space to the digital component (YUV/YC_RC_B) color-space. In addition to RGB to YUV transcoders, there are also YUV to VGA transcoders and many other types.

Transcoding devices fall into two sub-categories, those with no scaling and those which scale the input to 480p. Units which provide no scaling face the issue of the digital TV not supporting the low resolution of 240p, even if it is in the correct color space. These devices are low-cost products from China and have many different names despite being the same exact product. An ebay search of “SCART to YUV transcoders” yields many product results. SCART to YUV transcoders are typically used to provide RGB support on a CRT monitor which only supports YUV. For reference, a pure transcoder is shown in Figure 4 below. These devices will receive little attention here given that they have such limited compatibility with digital TV sets, instead we will focus on devices which are supported by most digital TV sets.

Figure 4: RGB to YUV transcoder, courtesy of SPECIALTYAV.

Units which provide scaling in addition to transcoding usually support a maximum output resolution of 480p. This ensures compatibility with most digital TV sets. A few notable examples of such devices are the HDBox Pro and the Arcadeforge SLG IN A BOX. However, these units treat the 240p signal as a 480i signal; while transcoding is accomplished, the picture quality is ruined by the de-interlacing. Although the artifacts introduced by deinterlacing of 240p are undesirable, the ability to use RGB as the source video makes these devices considered worthwhile. This is because TVs apply deinterlacing to the composite video input, so regardless of whether you use these devices or not the video signal is being deinterlaced. Transcoders typically introduce a noticeable amount of latency to the video processing data path which is perceived as input lag by the user.

Given that transcoders usually occupy a lower price bracket than line doublers and full scalers, it makes sense to use them in certain situations. If you are playing a slower-paced game where user input timing is not as critical then processing latency is acceptable. If the device is intended to be a one-off solution to provide support for a single console, then the robust solution provided by a full scaler is not needed. Our project at the minimum will support signal transcoding in order to be compatible with all of the color-spaces used in older video game consoles.

3.1.2 Line Doublers

The next set of devices scales the input video signal in a way that is much more pleasing to the eye than the methods used by previously mentioned devices. This scaling method is called line doubling which essentially duplicates each line. Here the signal is treated properly as 240p and not 480i; therefore no deinterlacing is applied in either the TV's processing stage or the external device's processing stage. This method leads to an increase in sharpness and brightness, both of which are desirable for 240p console art styles. Such devices are referred to as line doublers. We include devices which are pure line doublers and devices which are full scalers with a line doubling mode.

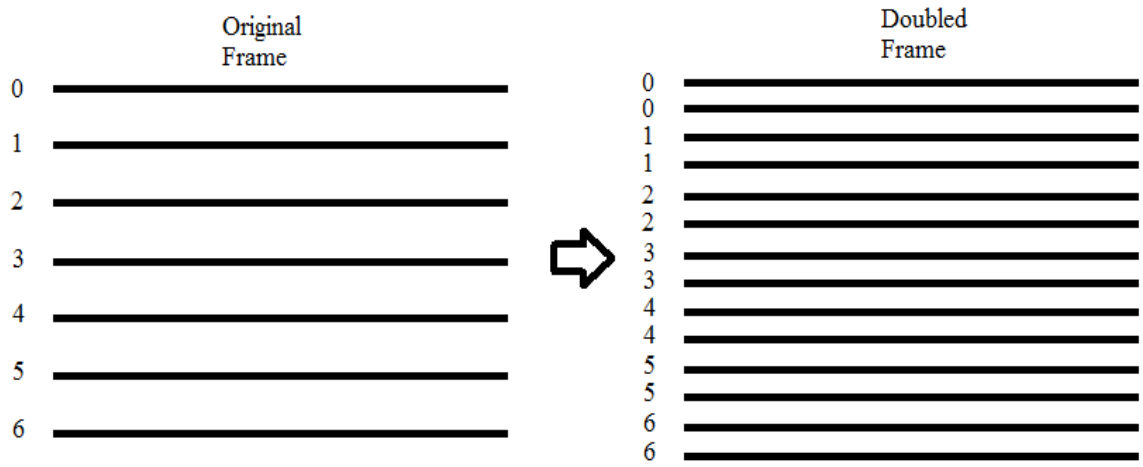


Figure 5: Basic idea of line doubling.

While digital TVs have trouble processing 240p video signals natively, most have little issues with the handling of 480p content. In certain situations a device which only performs a scaling of a factor of two can be ideal, such scaling can be done in a manner which introduces minimal input latency. However, if the TV has poor 480p handling then a full scaler may be necessary. Line doublers use a fast scaling method which is two-pronged, first horizontal scaling is performed with vertical scaling following. Horizontal scaling is accomplished by using an FPGA pixel clock which is twice the frequency input video signal's pixel clock, the pixel clock determines when to send a new pixel to screen. Since a picture is scanned to the screen in rows of pixels doubling of the pixel clock frequency doubles the pixel width of a picture row. To accomplish vertical scaling, each picture row is stored in a buffer and then when then each buffer is read twice to the screen. This double the height of the image by repeating every horizontal row twice. This scaling method requires very little processing time as the only active operation is the buffering of a single picture line. In comparison, other scaling methods rely on a partial to full frame buffer with costly multiply-accumulate operations to output the scaled frame.

Regarded as one of the best 240p line doublers on the market, Micomsoft's XRGB-3 is a full scaler with a special mode, B1, which lets it act as a pure line doubler. In this mode the XRGB-3 supports scaling of 240p to 480p with various analog video inputs from composite to scart. The XRGB-3 is able to provide near latency-free scaling by using a VGA output, no conversion to digital video is performed. This means you can use the XRGB-3 with a progressive scan CRT you can enjoy a full bright 480p image from a 240p console. However since the signal output is VGA, the user will still experience input latency as the XRGB-3's analog output is converted to a digital format by the TV. The XRGB-3 does not support 240p over component, only RGB, therefore a RGB to YUV transcoder is still needed for 240p component video support. This is only a slight issue as many video game consoles support both component and RGB video output.

Line doublers commonly support a feature known as scanline emulation. Scanline emulation darkens every other picture line in order to mimic the way a 240p video signal

is scanned to a CRT display. Scanline emulation is desired by many users of video scaling devices as it helps achieve an image that is similar to the classic experience of playing the video game on a CRT. Our device will at minimum support line doubling and color-space transcoding.

3.1.3 Full Scalers

Representing the peak class of devices in video processing devices for game consoles are full scalers. Full scalers are devices which can scale the input video resolution to an arbitrary output resolution, though realistically the output resolution is limited to what current TVs support. Here we will consider devices which support an output of 480p, 720p, and 1080p. Image scaling in such devices in accomplished through various scaling algorithms, manufacturers do not often disclose their algorithm, leaving it to the users to determine by inspection the algorithm used. Typically an ASIC is used as the primary scaling hardware, though FPGAs have started to appear with increased frequency lately.

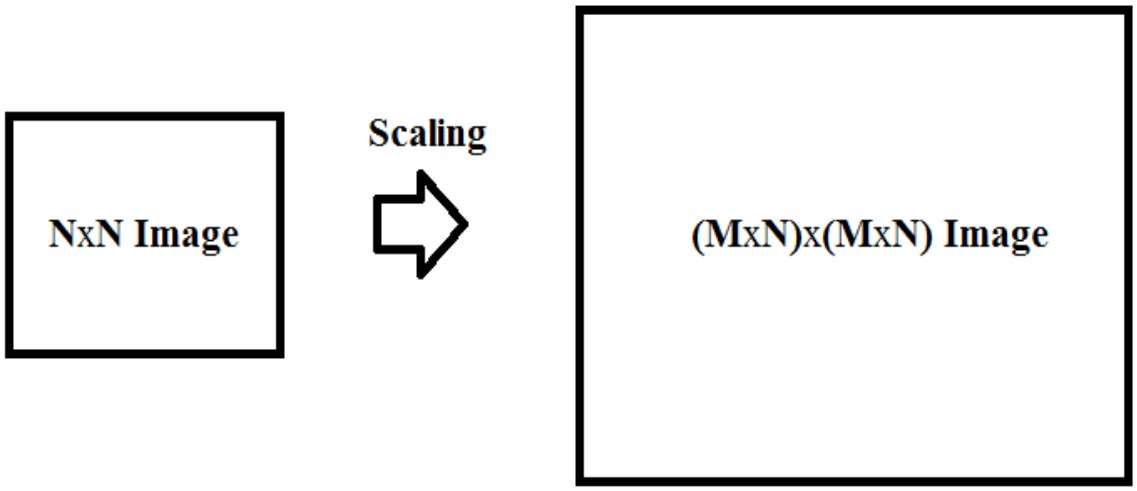


Figure 6: Flexibility provided by full scalers allows the user to choose an arbitrary output resolution within device limitations.

The previously mentioned XRGB-3 features a full scaler mode in addition to its line doubling mode. The XRGB-3 supports output resolutions up to 1080p although its scaling method requires fine-tuning and adds more input latency than other devices. The current best-in-class device is another product from Micomsoft's lineup, the XRGB-mini Framemeister. At the price of $350 the XRGB-mini does not come cheap. The Framemeister supports emulated scanlines, 1080p, and many other image adjustment options. As far as video inputs, the Framemeister supports component, composite, s-video, HDMI, and SCART. The Framemeister is very competitive in regards to input latency, with an added delay of only 28ms. Unfortunately all scaling is done by the Framemeister's full scaling chip with no option to enter a dedicated line double mode. This means that even if the user only needs 2x scaling, which should incur minimal delay, the signal still has to spend 28ms being processing in full scaling mode.

Our device sits roughly in-between the line doubling and full scaler classes of device. It will support multiple analog video formats at 240p as input with a maximum output of 720p. Our device will include a dedicated low-latency line doubling mode in addition to emulated scanlines. It is therefore fitting that our device be known as the “Super Doubler”.

3.2 Relevant Standards

Television is technically defined as the conversion of light to electrical energy in order to share information. With a variety of electronic video equipment involved in this process, standards and protocols had to be developed in order to monitor how this information was being created and exchanged and to ensure consistency. The justification for our project design plays off of the existing disparity in picture quality when modern digital televisions are interfaced with retro peripherals, such as Nintendo, Genesis (Sega) and PlayStation. When these older entertainment consoles were first created, they were designed to be compatible with analog televisions and output at 240p. When the digital LCD and plasma televisions were designed, they did not yield quality images or performance when run with the older consoles. This dissemblance is largely due to standards set up by the committees under the umbrella organization ISO, or International Organization of Standards. The ISO is a worldwide organization consisting of over 100 countries that moderates technical standards for science, technology and economic measures. There are many organizations around the world that fall under the ISO that regulate video including the National Television System Committee (NTSC), The Advanced Television Systems Committee standards (ATSC), Sequential Color with Memory (SECAM), and Phase Alternating Line PAL/SCAM.

The NTSC, used mostly in North America, was first created in the 1940’s to set forth the minimal requirements for analog video in black and white. When color television was created, in 1953 the NTSC revitalized the original standards for analog video to include color information; referred to Conventional Definition Television (CDTV). These standards were continued up until 2010 when digital video began to become more popular and analog transmitters were forced to shut down. The ATSC was started in the 1980s when the digital video first came about and revitalized the outdated analog video protocols to create criteria standards for digital video, including SDTV, EDTV, and HDTV. Whether referring to digital or analog video standards, several components are referenced such as image resolution, aspect ratio, pixel aspect ratio, scanning process, audio frequency and frame rate.

Table 2: CDTV standards.

CDTV Standards	Pixel/line	Aspect Ratio	Line Count	Frame Rate	Scan Mode
NTSC	640	4X3	480	29.97	Interlace
PAL	760	4X3	580	25	Interlace
SECAM	760	4X3	580	25	Interlace

Table 3: ATSC Digital video standards.

Resolution		Aspect Ratio	Pixel aspect ratio	Scanning	Frame rate (Hz)
Vertical	Horizontal				
1080	1920	16:09	1:01	Progressive	24 30 25
				Progressive	60 50
				Interlaced	29.97 30 25
1080	1440	16:09	HDV (4:3)	Progressive	24 30 25
				Progressive	60 50
				Interlaced	29.97 30 25
720	1280	16:09	1:01	Progressive	24 30 60 50
480	720	4:3 or 16:9	SMPTE 259M (10:11 OR 40:30)	Progressive	24 30 60 50
				Interlaced	29.97 30 25
480	640	4:03	1:01	Progressive	24 30 60 50
				Interlaced	29.97 30 25
480	528	4:03	(40:33)	Progressive	25
				Interlaced	29.97 25
240	352	4:03	(10:11)	Progressive	25

Image resolution refers to the quality of the video image, which is controlled by the number of pixels in a horizontal scan line multiplied by the number of scan lines in a frame. The resolution of a picture refers to how close the horizontal lines can be to each other and still be visible. The relevant standard for analog NTSC video was 480 horizontal lines comprised of 640 pixels per line. This gives us the spatial density resolution, or the number of pixels that make up one frame and it is usually denoted by (640X480). The ATSC standard for digital video was increased to 720X486. When the number of pixels is increased the image becomes more detailed. The higher the resolution of an image the crisper the image and can be displayed on a larger screen while maintaining the integrity of image.

The next standard is the pixel aspect ratio, which refers to the size and shape of the individual pixel. When original analog systems were set to become digitalized, the number of scan lines could not be changed due to the original standards set forth by the National Television System Committee (NTSC). In order to change the pixel aspect ratio, the number of pixels per scan line where increased which changed the shape of the pixels to a narrow, vertical and rectangular shape. The amount of pixels within an image directly correlates to the amount of information within that image. When you increase the number of pixels per line you add to the image resolution. Current digital standards for a 480p system are 4X3 or 1.33X1 ratio. This poses a problem when running a 240p program or game that uses the original standards set forth by the NTSC at a pixel ratio of 0.91:1; creating a disparity in image resolution.

Scan modes is the next standard reviewed and there consists two different types. The first is interlace scanning, which splits each frame of a video into two fields which in turn can increase the frame rate of the video image without increasing the bandwidth. One of the fields contains the odd lines of information and the other contains the even lines. When the video is played the two fields are “interlaced” together to form one frame. The result of this type of scanning produces a better quality image and reduces the "flicker" of the image. The benefit of this type of scanning is that it reduces cost but decreasing the amount of bandwidth. In addition, this scan mode provides a higher refresh rate which makes objects that are in motion in the image appear more smooth. With progressively scanned images, the frame is not split and instead is scanned from top to bottom as well as each line in the frame is scanned and then reproduced as one frame. So essentially each line is redrawn in sequence. Modern digital televisions run on progressive scanning, hence the “p” in 480p. When a 240p video console is utilized, the current standard equipment on a modern television is not equipped to convert the 240p to 480p and usually incorrectly converts from progressive scanning to interlacing scanning which distorts the image and creates processing latency.

The final standard is the frame rate. The frame rate refers to the number of full frames that are scanned per second and basically determines the speed in which an image is scanned. The standard frame rate for a 480p can be 24, 30 or 60 frames per second. If the scan mode is progressive or interlaced, it will be denoted as a p or i behind the number. 24p and 30p are progressive formats that are used mostly in translating a video signal to film where as 60p is utilized frequently with HDTV. For the analog video the standard is 29.97 fps.

As you can see, these standards set forth for both analog and digital video contribute to difficulties with connecting older entertainment peripherals to newer digital technology. When using a 240p retro video game console that originally processes analog signal, with a HDTV that may lack analog inputs can distort the image and create lag time when trying to play the older games. The video conversion device that we have designed will be an intermediary between the two systems, almost like a bridge, that with convert the analog to HMDI and at the same time upscale the video resolution allowing the user to play older games with a newer television.

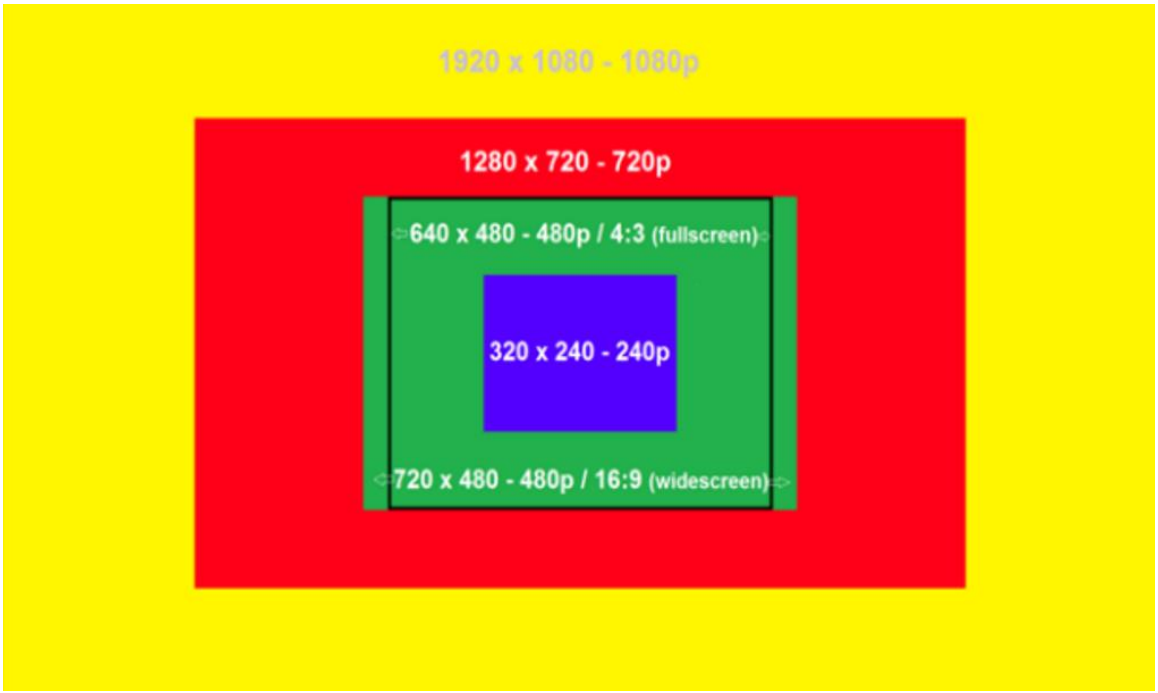


Figure 7: Video screen size in relation to resolution. Courtesy of RetroRGB.com.

3.3 Realistic Design Constraints

There can be several constraints that are intrinsic to any product design including economic, social, manufacturability and sustainability. These restrictions or hindrances can be roadblocks to design and development of a product or can spur creative problem solving that guides the conceptualization of a new and unique product. At this point in our project it is difficult to ascertain all design constraints as the prototype has not yet been completed but being proactive in identifying obvious constraints will help to avoid problems during the next phase of the project. Within this section of the paper we will discuss the impact of certain constraints on the video industry itself and how that guided our decision making for the design of our video conversion unit, The Super-Doubler.

What we propose to do with our video conversion unit is to create an intermediary device between classic 8-16 bit analog video game consoles and modern digital televisions. This device will allow for a high resolution image with little to no lag time as well as provide the capabilities for the player to adjust image properties and control scan lines with a simple device that is also more cost effective compared to similar products on the market today.

The health and safety of the consumer utilizing our product is of the upmost concern and is a key factor when determining the design of our system. In this day an age where litigation is very common, it is important that all foreseeable hazards be eliminated and the consumer notified of potential harm regarding certain aspects of design than cannot be mitigated.

The main concern in regards to the health and safety of the consumer is a responsible and respectful utilization of electrical components. Our device requires the consumer to have to connect several connectors to electrical devices such the TV, video game console and the Super-Doubler. Incorrect connections or removing connections when the device is on can potentially result in harm to the user or the machinery. Our responsibility resides in a component design that limits the consumer exposure to electrical shock and providing disclaimers to the consumer regarding potential harm from the misuse of the device and its components and connections. As with an electrical device, caution should always be exercised. In addition, many times devices can heat up quickly when on. In regards to design, we would research and choose a component shell and insulation to limit overheating as well as provide clear communication to the consumer regarding heating up the device and its hardware.

A second aspect in regards to health and safety is the safety of children who may use this device or who may come in contact with the small pieces and cords that can present a choking hazard. Ensuring that our product utilizes tight locking connections and smaller cord lengths and limiting small removable hardware will limit this constraint however as with other safety hazards, it is important to communicate with the consumer the potential harm and to strongly recommend that it be kept out of reach of small children.
In addition to health and safety constraints there are also environmental constraints to be considered into the design of our product. Out product include microchips and electrical circuitry that functional optimally under certain temperatures and environments. In the structure if our device, it will be imperative to creative a tight locking seal to prevent moisture and dust from entering and corrupting the computer and electrical components. In addition, it will be important to communicate optimal temperature ranges to the consumer based off of the data sheet information for the microchips and other hardware to the consumer. Extreme heat and cold temperatures are to be avoided.

The next constraint to be considered is cost effectiveness. Currently, with the high competitive environment, most manufacturers of modern digital televisions and HDTVs outfit their products with the bare minimum requirements for analog connectors and support, which often turns out to be composite, in order to save money. With the change in the standards from analog video (NTSC) to digital (ATSC), AV equipment that utilizes the outdated analog signal is becoming obsolete. It is the bottom line that drives most developers in the video industry to provide top of the line technology with competitive pricing that yields a profit and yet costs minimal to manufacture. Products on the market today that claim similar results to our product design are pricy and can cost the consumer upward of $400 for a video conversion unit to play their classic video games. These expensive alternatives do not afford the gamer the capabilities to adjust the image for the classic video game image and effect. The design of our product would have a large impact on the economics of the video industry. With an affordable conversion unit on the market, television manufacturers would not be forced to add additional and often costly additions to current models to reach the retro gamer consumer base, thus affecting the amount of units sold.

A cost effective design is what drove some of the decisions that we made in regards to product design in order to decrease the cost responsible for by the consumer. Even the best ideas can come to a stop if they become difficult or costly to manufacture or mass-produce. This is a vital component to the marketing value of our device. There are other devices out there that are similar and can recreate similar effects to our product design. The mark that sets our product above the others is that not only is it cost effective for the consumer but it is also simple and relatively cheap to manufacture. Initial budget estimate can be seen earlier in this document for a more detailed outlook. One of the difficulties approached by the group is that parts for our device are often sold in bulk. While this is a plus when mass producing technology equipment, as buying in bulk is normally cheaper, it created a higher than expected initial cost. One factor we took into consideration is the option to have more components than we need when creating our prototype in order to account for part failure as well as the ability to create multiple prototypes if desired. Ultimately our ambition is to cut back on cost once the prototype has been created and the necessary parts are identified. By utilizing bulk buying for hardware pieces as well as simplifying the design will allow us to create an effective product that will not put a financial strain on manufacturers and the consumer and allow us to be competitive with current technologies.

Another design constraint is the sustainability of our product. As technology continues to advance, older peripheral technologies become obsolete, as they are often unusable as video standards continue to change. With the design of our device, older technologies would be more sustainable over time as they would be able to connect with current and even newer technologies with small adjustments made. This also lends to the sustainability of the video conversion unit we are creating, as there will always be a need for a device that bridges the old with the new.

Video gaming is a social phenomenon that has shown recent upward trends in popularity. Current statistics report an increase in the number of people playing video games. At present the largest portion of gamers are in the age group 36 years and up at 36%. Within that group, 25% are older than 50. There is a rise in gaming in the older age range, which first started gaming with classic consoles such as NES and SEGA. This is a large consumer base, that could be targeted and would benefit from an affordable and dynamic video game conversion unit, such as the one we are creating.

Identified above are the realistic design constraints that we have faced thus far in our project and of which have influenced our decision making process in regards to the procedures and architecture of our product. Undoubtedly, as we progress further into the design and actual prototype development of our conversion unit, more design constraints may become apparent and will need to be addressed.

3.4 Technologies

In this section we will discuss the technologies and components of video and audio that are relative to our prototype including: relevant research and impact on design, pros and cons of specific devices and components, and rational behind our design decisions in regards to specific hardware chosen.

3.4.1 Data Signals

The higher density of pixels allows HDTVs to display a more detailed picture, but should you use one of your favorite retro video game, for instance the super Nintendo, which outputs right about 57,344 pixels, you would soon realize that either the TV is not working properly or the video game resolution image needs drastic improvement and upscaling. Adopting newer technology usually means that you either have to give up on your older video games or settle for a subpar image with associated lag time in performance. While some may be content with this outcome, there are many that require a better solution and this is where our proposed conversion device comes into play. Before specific design components can be discussed, there needs to be a general understanding of why there is a disparity between older analog video devices and the newer digital technologies. Both analog video and digital video and their components will be discussed in detail in this section of the paper.

3.4.1.1 Analog Video

Analog video is the first video recording method that stored red, blue and green color waves. Essentially an electrical signal was recorded on a magnetic tape (VHS) which may or may not be an exact replica of the original image. Analog video begins with the capture of the images by a camera. The camera scans the video lines one by one from top to bottom until the whole video field, in other words the whole view in which the camera is capturing the images, are processed. With analog video the number of lines are fixed and the amount of detail of an image is largely decided by the frequency response. The camera then re-

scans the lines again from top to bottom. To sync the two systems properly, the camera transmits extra sync pulses so that the TV set can synchronize at the same line, which is transmitted, and at the same field. If the sync pulses are not received properly by the TV set because the video signal is too weak, then the screen starts scrolling and flipping and you get a corrupted image. One of the drawbacks of analog video is that they are susceptible to corruption and breakdown as they are not protected as digital data is. Also analog video signals are subject to noise interference which can reduce the accuracy of the signal and the quality of the image. One of the perks of analog video is that it consumes less bandwidth but information is easily lost or distorted.

The higher density of pixels allows HDTVs to display a more detailed picture, but should you use one of your favorite retro video game, for instance the super Nintendo, which outputs right about 57,344 pixels, you would soon realize that either the TV is not working properly or the video game resolution image needs drastic improvement and upscaling. Adopting newer technology usually means that you either have to give up on your older video games or settle for a subpar image with associated lag time in performance. While some may be content with this outcome, there are many that require a better solution and this is where our proposed conversion device saves the day. Before specific design component can be discussed, there needs to be a general understanding of why there is a disparity between older analog video devices and the newer digital technologies. Below we will discuss different data signals.

The understanding of how analog video signal works is indeed fundamental to the end goal of our project as most of the project will be dealing with analog signals and the conversion of its signals so that the creation of the "bridge" between the "older technology" and newer one can be made possible end goal of high quality image output. There are several different standards used worldwide in relation to analog video including PAL (Phase Alternation by Line), NTSC (National Television System Committee) and SECAM (Systeme Electronique Couleur Avec Memoire); though PAL and NTSC are used most often. Below the basic concepts of analog video and its applications will be discussed.

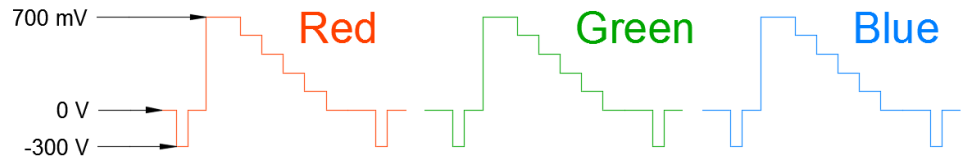
Analog video in a nutshell begins with the capture of the images by a camera. The camera scans the video lines one by one from top to bottom until the whole video field, in other words the whole view in which the camera is capturing the images, are processed. The camera then re-scans the lines again from top to bottom. To sync the two systems properly, the camera transmits extra sync pulses so that the TV set can synchronize at the same line, which is transmitted, and at the same field. If the sync pulses are not received properly by the TV set because the video signal is too weak for example, the screen starts scrolling and flipping and you get a corrupted image.

3.4.1.1.1 Analog Synchronization Signaling

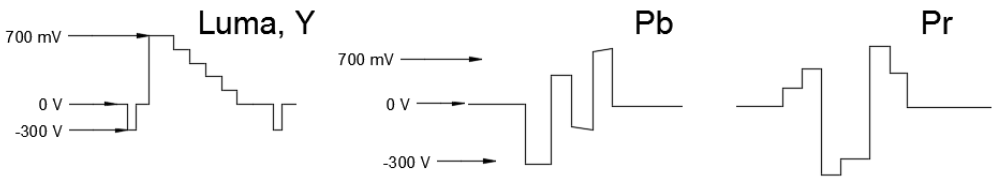
Analog video requires an extra signal at the end of each scan line and frame to ensure that when the video is transmitted it can be correctly refigured on the TV screen. Horizontal sync pulse separates the scan lines and denotes when a new line starts. Vertical sync is a longer pulse when compared to horizontal sync and indicates to the receiver when a new field is to be started.

Component video synchronization may be sent in different ways since it requires an extra synchronization signal to be sent along with the video. There a several different types of sync including, separate sync, composite sync, sync-on-green, sync-on-luminance, and sync-on-composite. Separate sync uses two separate wires for horizontal and vertical synchronization and sends 5 different signals for red, green, blue and horizontal/vertical sync. Composite sync combines horizontal and vertical synchronization onto one pair of wires. Sync on green, combines composite sync with the green signal in RGB and only 3 signals are sent. Sync-on-luminance combines the luminance signal (Y) if the YPbPr system. Finally, sync-on-composite utilizes a composite video signal and combines it with RGB and utilizes the SCART connector. This type of sync is used mostly for devices that cannot process the RGB signals. The drawing below depicts the different synchronization signals for RGB, component, S-video and composite video.

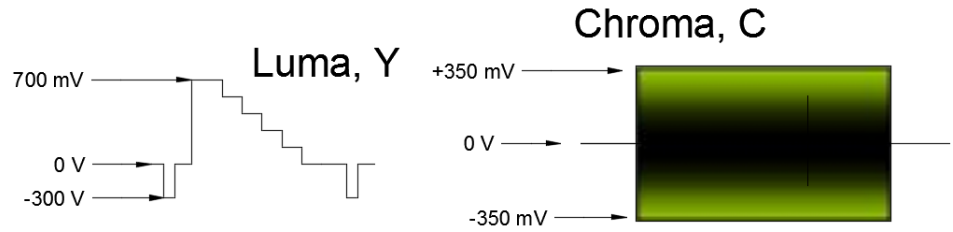
Interestingly enough there are can also by sync issues between the audio and video signals during conversion creating a timing issue between the video and sound. This issue can result in lip syncing issues in which the image of a person talking does not line up with the speed of the sound. The standard sample rate for music is 44.1 kHz when coupled with anti-aliasing filters and converters. It was found that this lower sampling rate, compared to 48 kHz, was more beneficial in that it provided the same quality sound but also allowed for more music to be processed with fewer issues. The standard sampling rate for video was determined at 48 kHz which creates issues when utilized with the lower audio rate and mismatched sample rates can contribute to the lag between video and audio.



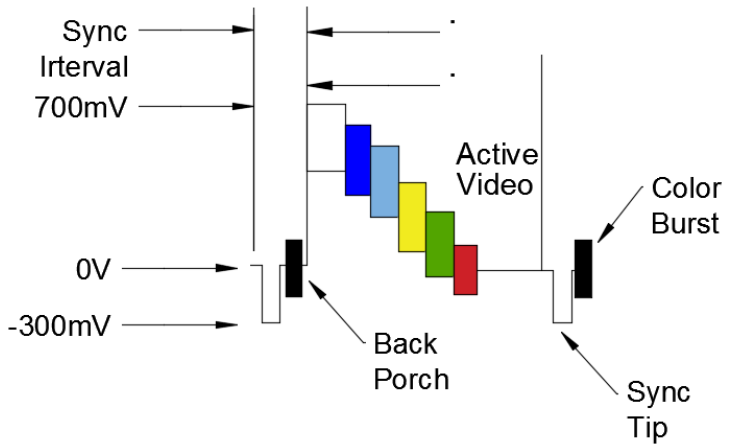
(a)



(b)



(c)



(d)

Figure 8: (a) RGB, (b) Component, (c) S-Video, (d) Composite video. Courtesy of maximintegrated.com.

3.4.1.2 Analog Audio

There are two basic types of audio recordings; analog and digital. In this section we present the analog audio and why it will be mentioned is this part of the project and its contributing use. Analog audio was used exclusively in early years as the main processing system for sound until digital devices became the standard. Analog audio uses a method in which the analog signals are recorded and it reproduces an identical copy of the original sound waves. Among the disadvantages of analog signals, its drawbacks can affect the quality of the ending sound especially if what the application is to convert analog audio in digital audio. Analog audio signals are susceptible to noise and distortion due to the innate characteristics of electronic circuits and associated devices. In order to adjust the analog audio waves before its conversion we will be using a low pass filter in which it will be cleaning up analog signals before conversion and processing. The THS7353 low pass filter is a perfect choice for the application that it will be used since it is compatible with all video buffer applications; the THS7353 is a lower-power I^2C configurable which makes it flexible for the application that it will have to achieve, further details of the I^2C configurations will be discussed in a later section. Since the audio signal encompasses a much smaller range of spectrum space than video, audio recordings in analog are direct recordings. To conclude we know that analog audio is important for our project in order to convert it to digital so that in the end we can carry it over HDMI for a single connection. Moreover we will be able to apply these concepts to our device to improve classic video game console output.

3.4.1.3 Digital Audio

Digital audio systems are an important aspect of our project in that it deals with the transition of digital equipment from analog environment sources. The achievement of this task requires precise steps in order that the analog signals are converted to digital audio system without disturbances and error. Digital audio systems utilize numbers to represent the original audio waveform as binary data, making the process known as pulse code modulation (PCM). PCM is the form of binary coding used in virtually all digital audio systems. Timing in this process is imperative in order to achieve stability of the audio output. If the instability is not addressed properly prior to this process, timing inaccuracies cannot be removed at a later time and also will result in an unstable stereo imaging and noise issues. So for accuracy the highest audio frequency that a digital system can encode must be less than or equal to ½ the sampling frequency. The low pass filter is a technology used to record, store, generate and manipulate sound using audio signals encoded in serial digital streaming. For our project we are using the Inter-IC sound chip (I^2S). The I^2S is a serial bus that that will connect the audio decoder to the HDMI receiver and disseminate PCM audio between the integrated circuits. Digital audio is useful in the recording, manipulation, mass production and distribution of sound. There are many benefits when using this system since it does not have the same noise audio problem that analog audio has.

Disturbances in a digital system do not result errors as in analog audio unless the disturbance is so great that the symbol is being misinterpreted as another symbol or disturbance in the sequence of symbols. Also, digital audio can produce a greater frequency of range, which may include but not limited to, compression, storage, processing and transmission components. Whenever you are dealing with corrections in digital audio signals, encoding may be applied for correction measures that might occur in the storage or transmission of the signal. Channel coding refers to a process that is completed when a signal is transmitted and received and it involves designating extra bits and absorbing extra bandwidth to allow for better communication.

3.4.1.4 Digital Video

Essentially the process of digitizing video is capturing the analog sine wave at different time span and encapsulating it in a numerical value for each measurement as time great than zero. In order to a more accurate digital reproduction the data has to be captured more often since the sine wave curve is changing continuously as time is progressing. Once data has been digitalized the information is sent in one digital stream that contains both video and audio.

The amount of data on that stream determines the quality of the image, however a stream with high quantity of data requires a larger bandwidth to transmit it. If a data stream has a large amount of details, but the bandwidth is too small the image quality will be poor. The information in the digital stream is in the form of zeroes and ones and it is organized in frames. From this point on, the frames are organized into packets and the packets are organized in segments. The different groups within a digital stream are encoded so that the receiving end can correctly organize it. A digital may contain information from more than one source so it is fundamental that the data from the frames, packets, and segments are sent and received correctly otherwise the digital stream cannot be decoded and the image will be of poor quality. The image drawn shown below gives a detailed visual demonstration as to the process of encoding and decoding an analog signal to digital.

The advantages of having a digital world where most information is shared digitally is that it is easier to manipulate and provides for a much faster editing. With a digital video signal, a person has the capabilities to separate the sound from the image where as in analog form both are congruent and not easily edited. Additionally, with digital video, the image and information is protected and not as susceptible to wear and tear as a video tape cane be. With digital video, the information can be copied multiple times with out losing information or compromising the integrity of the video. With the digital video the data files can be very large and unmanageable without digital video compression.

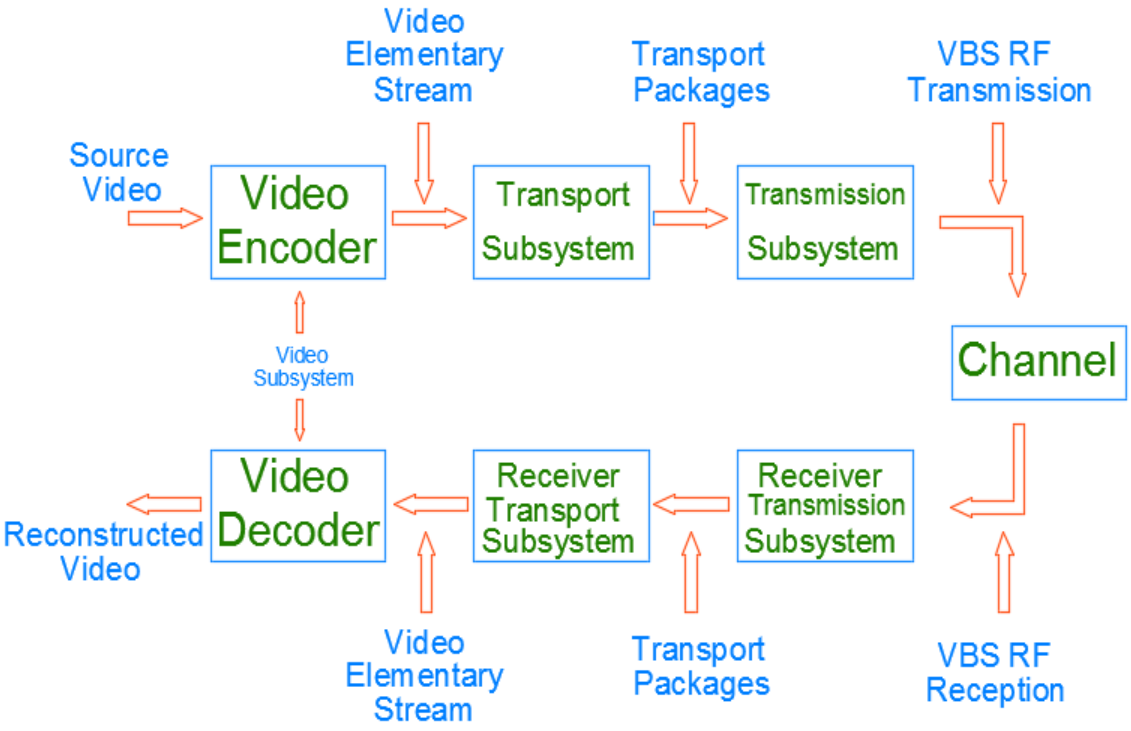


Figure 9: Encoding and Decoding an Analog Data Signal.

3.4.1.5 HDMI

In an effort to provide a low-noise output video signal which is compatible with many displays, we chose to use HDMI as our system's primary output. HDMI uses the EIA/CEA-861 video standard and allows for the transmission of uncompressed audio and video signals using a single cable. An HDMI link can be used for RGB 4:4:4, YC_BC_R 4:4:4, and YC_BC_R 4:2:2 pixel encodings. With a maximum pixel clock of 340MHz, HDMI 1.3 supports resolution up to 2560x1600 using a single digital link. Data is sent in three packet types, control, data island, and video data. The video packet contains pixel information and the data island packet contains audio information as well as auxiliary information. A special encoding scheme is used which reduces the number of transmission line voltage changes by representing an 8-bit packet in an expanded 10-bit form.

A timing diagram for an entire 720x480 pixel video frame is given in Figure 10 below. Video data packets are sent during the active pixel region, this constitutes the color information shown on the display screen. The audio and control packets are sent during the blanking periods before the active regions. A control period is required in between any two periods which are not control periods. Table 4 describes the data transmitted and the encoding type used in each transmission period.

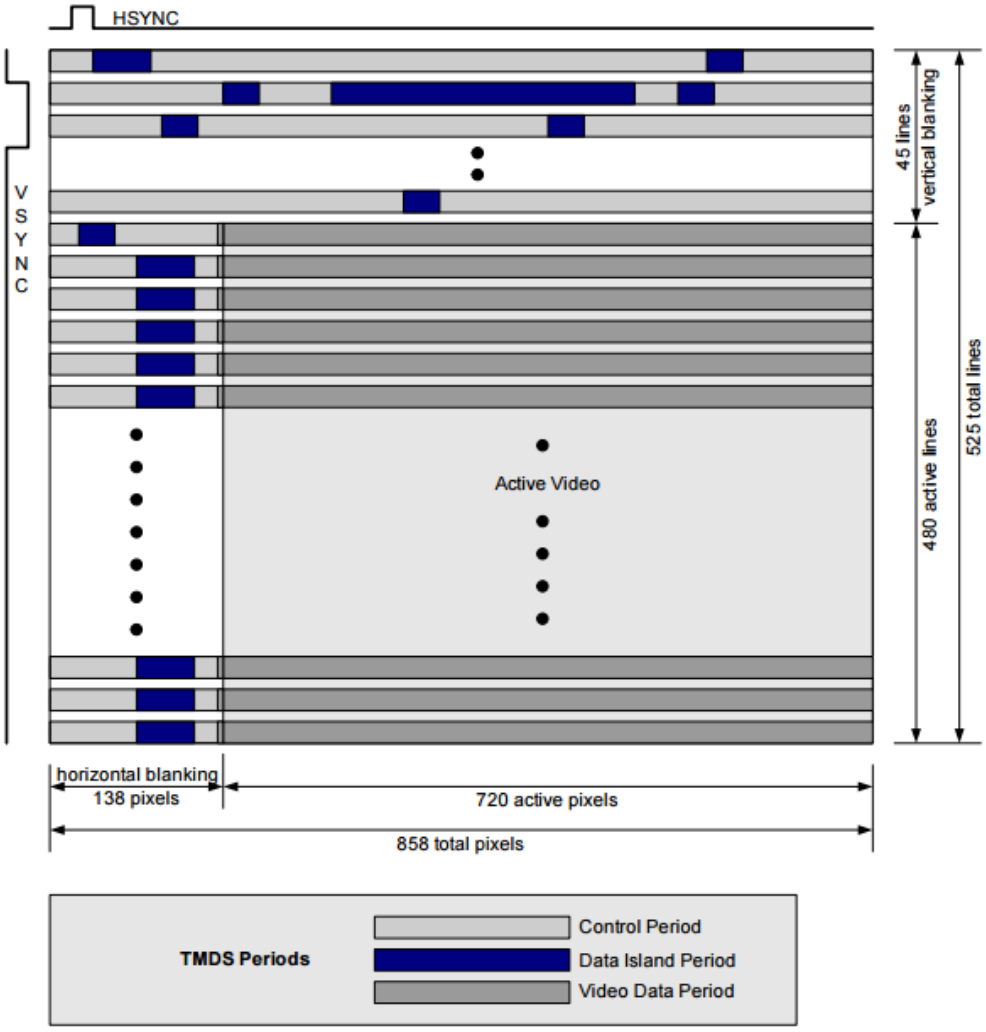


Figure 10: Timing diagram for HDMI video frame. HDMI permission pending.

Table 4: Content transmitted during various periods and the encoding type used.

Period	Data Transmitted	Encoding Type
Video Data	Video Pixels	Video Data Coding (8 bits converted to 10 bits)
Data Island	Packet Data -Audio Samples -InfoFrames HSYNC, VSYNC	TERC4 Coding (4 bits converted to 10 bits)
Control	Control -Preamble -HSYNC, VSYNC	Control Period Coding (2 bits converted to 10 bits)

The HDMI standard utilizes a signal format known as transmission minimized differential signaling (TMDS) which helps to reduce noise and improve signal bandwidth. A large benefit of TMDS is the ability to send audio, video, and data in the same stream as packets. The term transmission minimized comes from the encoding scheme used in a TMDS link; the encoding reduces signal transitions from high to low and vice versa by representing 8-bit packets as 10-bits. Differential signals using twisted pair wiring helps to reduce noise due to electromagnetic interference. While an FPGA is capable of performing the 8-bit to 10-bit conversion as well as outputting the required differential signals, for our projects needs it is more practical to use a dedicated HDMI transmitter like the ADV7511. If we were to use the FPGA to create the TMDS HDMI signal it would also require a way for the FPGA to capture the digital audio data from the input audio decoder.

3.4.2 Communications

Arguably the most important aspect of this project not related to direct audiovisual manipulation is the communication amongst all of the discrete device components. As specially designed ICs are available to perform many of the required data transformations and service input/output needs, the primary challenge is then to ensure all of these devices are correctly synchronized and configured relative to one another for their specific task. This is accomplished by the use of one of several serial communications protocols.

For our purposes specifically related to inter-device communication (and thus ignoring in-system reprogramming addressed in section 3.4.5) the relevant protocol options are Inter-Integrated Circuit (I^2C), Serial Peripheral Interface (SPI), and Universal Asynchronous Receiver/Transmitter (UART). I^2C is required for use in this project as several of the special purpose ICs are controlled via this protocol. SPI and UART are both presented as

potential solutions for inter-device communication for the microcontroller and FPGA. These communications methods are treated in this section.

3.4.2.1 I²C (Inter-Integrated Circuit)

I²C is a synchronous serial communication protocol intended for use in communications between devices located in relatively close proximity to one another (i.e. for our project on the same PCB). I²C requires an extremely small hardware footprint utilizing just two wires for communications, SCL acting as the clock signal and SCA acting as the data wire. Both wires are pulled high and devices manipulate the bus by pulling the specified line low. Multiple devices can exist on the same I²C network as each device has its own address specified via 7 or 10 bits. These devices exist as a single Master, responsible for triggering the clock and initiating communications with other devices designated as Slaves. Conventional I²C implementations provided for only one master though a multi-master mode does exist where devices are able to change between master and slave as necessary. A diagram of a typical I²C network can be seen in Figure 11.

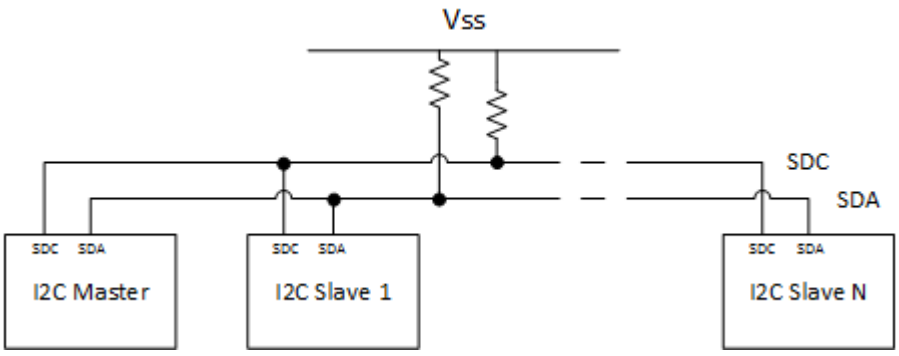


Figure 11: Diagram demonstrating a typical I²C network arrangement.

I²C data transfer requires use of two separate frames. Address frames specify the device targeted by the master for communications, and data frames carry the data for the requested communication. The address frame also specifies whether the master is requesting to read or write to the addressed slave. Following the address frame, the master or slave (depending on the communication mode selected in the address frame) manipulates the SDA line by leaving it high (logic 1) or pulling the line low (logic 0) to send data 1 bit at a time in 8 bit frames. The SDA line is sampled when SCL is high and changed (if necessary) when SCL is low. The standard specifies that addresses and data are both sent most significant bit first. Each frame ends with a single Ack/NoAck bit. The device that received the previous frame is responsible for pulling SDA low during this bit window to indicate that it successfully received the sent frame. The figure (Figure FIGURE NUMBER) shown below shows a diagram of sample I²C communication via an address frame and single data frame including all bits detailed previously. This communication is commonly supported at three speeds defined in modern I²C standards, 100 kHZ (100 kbps), 400 kHZ (400 kbps “fast mode”) and 1 MHz (1 Mbps “fast mode plus”).

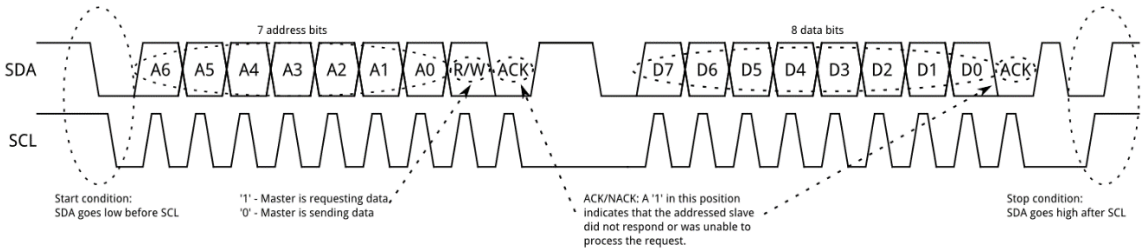


Figure 12: I^{2}C example address and data frame clock diagram. Courtesy of Sparkfun.

I^{2}C is supported by a large number of ICs requiring advanced configuration or serial communications, and supported specifically by the various decoder, encoder, and transmitters required in our project. Additionally, there are readily available I^{2}C implementations for FPGAs via OpenCores and support for I^{2}C at the hardware and software level in all considered microcontrollers.

3.4.2.2 Serial Peripheral Interface (SPI)

While I^{2}C will be used extensively in this project due to requirements for specific devices like the audio/video encoders, decoders, and transmitter/receivers, it is but one option for implementation of the microcontroller and FPGA communications interface. Another option, presented here, is Serial Peripheral Interface, commonly referred to as SPI. SPI shares many functional characteristics with I^{2}C. Both protocols are synchronous utilizing a clock signal to determine when to send, read, and receive data. Both also have this clock signal controlled by a single device designated as Master and support multiple devices configured as Slave.

There are two key differences between the protocols. One is that under SPI, the Slave currently selected for communications with the Master is selected by a separate signal line called Slave Select (SS). For our implementation there would be at most 2 SPI devices (the microcontroller and the FPGA SPI interface) so this is essentially an enable wire as opposed to I^{2}C's initiation via address frames. The second is that SPI supports full duplex communications so that Master and Slave can both send and receive data simultaneously if configuration permits. This is accomplished by using two separate data wires, Master Out Slave In (MOSI) and Master In Slave Out (MISO). The wire names accurately describe their use as each being responsible for communication in one direction. Figure 13 below shows the control path for all wires in an expected implementation within this project featuring a single master and single slave.

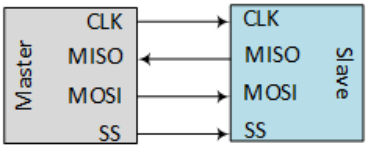


Figure 13: SPI control signal flow and connections expected in a simple implementation.

These additional connections are a key differentiator between SPI and both I^2C and UART communications. Where both I^2C and UART are implemented using only two wires, SPI requires 4 to support its greater feature set. At the implementation level, SPI is also made more complicated in that it is not a true communications standard. Communication rates and data formatting (frames, packets, etc) are all left to the end user to define their implementation. While this loose definition does provide flexibility, the additional work required to define a complete communications interface without even such building blocks as agreed upon frame sizes is potentially considerable.

While the standard itself is not particularly clearly defined, SPI support is implemented in most microcontrollers in the class needed for this project. Microcontroller vendors typically also provide libraries or reference designs utilizing and implementing the SPI bus. Also, SPI functional interface blocks are readily available for FPGAs via OpenCores.

3.4.2.3 UART (Universal Asynch. Receiver/Transmitter)

Universal Asynchronous Receiver/Transmitter communications, or UART, is conceptually the simplest of the protocols reviewed for this project. Like I^2C it uses only two wires for communication, but as the communication itself is asynchronous there is no clock wire. One wire is designated as the device's transmit (Tx) wire, and the other is the receive (Rx) wire. Each device then has its Tx terminal connected to the other device's Rx terminal and its Rx terminal connected to the other device's Tx terminal. A simple drawing in Figure 14 below shows this connection as expected between two devices.

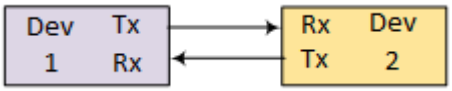


Figure 14: Two device UART connection.

Since UART does not use a clock wire, it requires explicit configuration prior to use concerning clock rates and frame formats. The data rate, number of start bits, number of data bits, number of stop bits, and parity format, and data endianness are all required to be known prior to correct communication. UART is supported by virtually all devices capable of serial communications whether as pure asynchronous serial or via a standard like RS232 or TTL. However, it boasts lower maximum throughput values than the other options, with data rates of around 115 kbps being a typical maximum before errors due to lack of throughput capabilities occur. Nonetheless, it poses an attractive alternative to the synchronous protocols if data throughput requirements between the microcontroller and FPGA can be kept low.

3.4.3 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. The HDL language of choice for this project is Verilog because of the language's low-level structure making it ideal for fine-grain image processing operations. The task of the FPGA in this system is to scale the resolution of each image frame in an input video stream by a positive scalar value. Multiple parallel data paths will be implemented to provide different processing modes, therefore a high-capacity FPGA is desired. In this section we discuss the various device families under consideration and the development environment chosen for this project.

3.4.3.1 Xilinx Spartan-3 Family

The first family of devices under consideration is Xilinx's Spartan-3 FPGA family. The Spartan-3 family is based on a 4-input LUT logic cell structure with capacities ranging from 1,728 logic cells for the low-end XC3S250 to 74,880 logic cells for the flagship XC3S5000 FPGA. Each FPGA in this family features hardware multipliers and digital clock management (DCM) blocks. The dedicated multipliers are a necessity for operations such as convolution which is frequently used in image processing. The digital clock management is needed to synchronized FPGA logic cells to external video clock signals, these DCM blocks help reduce clock skew and jitter.

The specific device in the Spartan-3 family under consideration is the XC3S400. At a cost of $33.75 per unit the XC3S400 comes equipped with 8,064 logic cells and 294kb block memory. The package type is a 256-LBGA with 173 available user I/O pins, 3.3V logic I/O is supported. The supply voltage is 1.14V-1.26V with a max operating temperature of 85°C. We would have preferred to use the XC3S1500 for its high logic capacity but due to low availability it was abandoned as a choice. The block RAM is arranged in 18kb blocks of dual-port RAM, dual-port RAM is useful to use as a frame buffer due to the ability to perform a read and write operation in the same clock cycle. There are 16 dedicated multipliers available, each is able to support two 18-bit operands. The XC3S400 comes equipped with 4 digital clock management blocks which will be needed to synchronize internal logic resources to external clock signals.

From a cost and availability perspective, the XC3S400 is a good choice. The device supports a sufficient number of I/O and has decent logic capacity. While the available memory is quite large, 294kb would may be inadequate for higher resolution frame buffers. A possible solution would be to utilize spare logic cells as distributed memory, however this should only be done as a last resort. The number of DCM blocks is also a cause for concern due to the number of external clocks that the FPGA will potentially have to synchronize with. The FPGA will need to synchronize with the video decoder's pixel clock, horizontal clock, and vertical clock in addition to the clock used for communication between FPGA and MCU.

3.4.3.2 Xilinx Spartan-6 Family

The second family of devices under consideration is Xilinx's Spartan-6 FPGA family. The Spartan-3 family is based on an improved 6-input LUT logic cell structure with capacities ranging from 3,840 logic cells for the low-end XC6SLX4 to 147,443 logic cells for the flagship XC6SLX150T FPGA. Each FPGA in this family features hardware multipliers and clock management tiles (CMT).

The specific device in the Spartan-6 family under consideration is the XC6SLX9. At a cost of $21.84 per unit the XC6SLX9 comes equipped with 9,152 logic cells and 589kb block memory, providing a much better price to capacity ratio than that of the XC3S400. The package type is a 324-LFBGA with 200 available user I/O pins, 3.3V logic I/O is supported. The supply voltage is 1.14V-1.26V with a max operating temperature of 85°C. Like the Spartan-3 family, the block RAM is arranged in 18kb blocks of dual-port RAM. New to the Spartan-6 family is the ability to treat each 18kb block as two independent 9kb blocks. The Spartan-6 family utilizes DSP48A1 slices which each contain an 18x18 multiplier, an adder, and an accumulator. The XC6SLX9 contains 16 DSP48A1 slices. The clock management tiles (CMT) expand upon the DCM concept with each tile containing two DCMs and one phase-locked loop (PLL). The XC6SLX9 contains two CMTs which puts it at four DCMs and two PLLs.

From a cost perspective, the XC6SLX9 is a better option than the XC3S400. The XC6SLX9 logic cells are much denser due to utilizing a 6-input LUT plus 4 D-flip-flop cell structure versus the XC3S400's 4-input LUT plus one d-flip-flop cell structure. While the available memory is quite large and almost double of that available in the XC3S400, 589kb would still be inadequate for higher resolution frame buffers. The XC6SLX9 has the same number of DCMs as the XC3S400, however it contains two extra PLLs will be useful in our design.

3.4.3.3 Xilinx Artix-7 Family

The final family of devices under consideration is Xilinx's Artix-7 FPGA family. The Artix-7 family is based on a 6-input LUT logic cell structure with capacities ranging from 16,640 logic cells for the low-end XC7A15T to 215,360 logic cells for the flagship XC7A200T FPGA. Each FPGA in this family features DSP48E1 slices and clock management tiles. Each DSP48E1 slice has a pre-adder, a 25x18 multiplier, an adder, and an accumulator. Each clock management tile has one mixed-mode clock manager (MMCM) and on phase-locked loop (PLL).

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 package type is a 256-LBGA with 170 available user I/O pins, 3.3V logic I/O is supported. The supply voltage is 0.95V-1.05V with a max operating temperature of 100°C. The block RAM is arranged in 36kb

blocks of dual-port RAM, dual-port RAM is useful to use 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.

From a cost and capacity perspective, the XC7A35T is a good choice. The available memory is quite large, 1800kb would be sufficient for our target resolutions. In addition to the 1800kb block memory, the XC7A35T allows access to up to 400kb of distributed memory by using spare logic slices. The XC7A35T comes equipped with five clock management tiles, providing fine grain control over multiple clocking signals. The FPGA will need to synchronize with multiple external clock signals, the phase-locked loops inside the CMTs will help facilitate this. The XC7A35T also features the Xilinx Analog to Digital Converter (XADC), which is an integrated ADC within the FPGA fabric.

Table 5: Comparison of chosen devices from various FPGA families.

	Spartan-3	Spartan-6	Artix-7
Logic Cell	8,064	9,152	33,280
Dedicated Multiplier	16	16 MAC	90 MAC
Clocking Resources	4 DCM	4 DCM, 2 PLL	5 CMT

Fortunately, there is a wide variety of FPGAs to choose from amongst the various device vendors. We narrowed our choices down to Xilinx FPGA devices due to familiarity with their design software. Given that the Spartan-3 family is getting older, it is becoming more difficult to find each chip in multiple packages. This narrowed our choices within Spartan-3 to the XC3S400. The XC3S400's 4-input LUT logic cell structure means that the resources are consumed quickly when dealing with complex logic functions, in comparison to higher input LUT devices. In addition, the XC3S400 lacks a sufficient amount of integrated memory required for image processing. The low number of hardware multipliers would also constrain the performance of any image processing algorithms implemented. The XC6SLX9 is more capable device than the XC3S400 at a much cheaper price point. However, the XC6SLX9 still presents a memory constraint and the low number of clock management resources might prove difficult during design. The XC7A35T is packed to the brim with resources which will be helpful in relieving design constraints. The large number of DSP48E1 slices will be useful in extracting the most performance out of real-time image

processing algorithms. The 1.8Mb of block memory will allow us to target higher resolution output modes. Finally, the five clock management tiles provides us with ample resources to manage our internal and external clocking signals.

The development environment used for this project is Xilinx's Vivado Design Suite. This software is required to configure the latest families of FPGAs in Xilinx's lineup which includes the XC7A35T. Xilinx simulation and design analysis tools will be used to exam details pertaining to device performance and functionality. The HDL language of choice is the Verilog language, chosen due to the designer's familiarity. Existing IP blocks in the Xilinx library will be used as needed, as will open-source Verilog modules.

To handle the physical prototyping of the FPGA subsystem, the ARTY development board from Digilent will be used. The board features the XC7A35T FPGA along with external DDR3 memory and oscillator circuits. The ARTY board uses a USB connection to program the XC7A35T FPGA. It should be noted that the ARTY lacks a video output connection. For initial testing a resistor ladder DAC will be used to drive a VGA output connection.

3.4.4 Microcontrollers

While an FPGA is ideal for the image processing subsystem, much of the control operations and interaction with subsystems outside of the audio/video processing subsystem are more readily accomplished with the assistance of a microcontroller. This microcontroller can specifically be used as an intermediary between user control and feedback – for example control requests coming from the remote control unit or on-enclosure buttons and status LEDs – and the underlying FPGA implemented processing structures. After surveying the various manufacturers the decision was made to limit consideration to two suppliers, Texas Instruments and STMicroelectronics due to the wealth of comprehensive documentation, community support, and development tools made available by each. Particularly in consideration of ARM microcontrollers, this restriction served to reduce our focus to a manageable subset of the enormous number of ARM offerings available.

3.4.4.1 Texas Instruments MSP430G2553

Outside of ARM based microcontrollers, the Texas Instruments MSP430G2553 was considered due to its continuing popularity. A mainstay of the embedded, lower-power world, the MSP430G2553 is a 16-bit MCU with a default clock rate of 16 MHz, 16 kB of program memory and 512 B of on-chip RAM. It supports serial communication over I2C, SPI, and UART. Texas Instruments provides an entire ecosystem of supporting software and hardware tools in the form of the Code Composer Studio and Energia IDEs paired with the MSP430GX2 Launchpad development board. The MSP430GX2 Launchpad provides a complete MSP430 development platform including a programmable user button, 2 user LEDs, on-board USB programming and debugging interface, and full I/O pin exposure. The Launchpad can also be used as a standalone programmer for MSP430G2xx series microcontrollers. All of this functionality comes at a low cost of approximately $2.00 for a G2553 IC and less than $10 for the Launchpad.

The MSP430G2553 is primarily considered due to its extremely low cost and previous team member experience with the chip itself, the Launchpad development board, and the Texas Instrument toolsets. The MSP430G2553 is also available in a 20-pin DIP package, easily paired with a solderable socket or hand soldered. The performance of the chip is greatly outpaced by the ARM offerings discussed below, though at the expense of cost, ease of use, and developer familiarity.

3.4.4.2 Texas Instruments TM4C123GH6PM

The lone ARM offering considered from Texas Instruments, the TM4C123GH6PM features an ARM Cortex-M4F core targeted for performance intensive signal processing applications. The 32-bit TM4C123GH6PM boasts impressive specifications with a clock speed of 80 MHz, 256 kB program memory, 32 kB on-chip data RAM, and 51 GPIOs. The TM4C123GH6PM features a single enormous benefit in developer use for our group in that Texas Instruments has available the Tiva C Launchpad series featuring the microcontroller. This Launchpad offering features all of the benefits found in the MSP430GX2 Launchpad but with a very capable ARM MCU. Another benefit to this part is the extensive Texas Instruments documentation and reference design material. This material is similar in style, organization, and contents to the accompanying MSP430 series documentation which the entire project group has previous experience with. This familiarity provides an early boost to design productivity. Texas Instruments resources and sample parts are also more readily available than many other manufacturers, even more so thanks to their partnership and funding with UCF in the Innovation Lab.

The TM4C123GH6PM is geared towards environments requiring large amounts of signal processing with special DSP instructions and enhanced floating point operation. While our project does feature uses for signal processing capabilities, we have elected to use dedicated ICs where those functions are required due to strict latency specifications. Part of the Cortex M4F chip's cost, at roughly $11 per single order ($9 per 100 count) is likely justified in many projects with this specialized functionality, but here represents an added cost for little projected benefit. Its significant performance metrics, while impressive and providing adequate headroom for microcontroller responsibility expansion in our project, also may be too powerful (and expensive) for our purposes.

3.4.4.3 STMicroelectronics STM32F0 Line

The STM32F030R8T6 and STM32F070RBT6 are two ARM Cortex M0 offerings from STMicroelectronics. The STM32F030R8T6 is part of STMicroelectronics' "Value Line" offering, comprised of ARM chips designed to compete in embedded spaces typically occupied by low power, low cost offerings like AVR devices. However, unlike AVR devices of the previously examined MSP430, Cortex M0 parts are capable of full 32-bit operation and have significantly more impressive performance characteristics. The STM32F070RBT6 and STM32F030R8T6 are so similar in specifications that they are examined together here.

Both chips utilize 32-bit data buses at 48 MHz in LQFP64 packages. The STM32F030R8T6 has 64 kB of program flash, 8 kB of on-device SRAM, supports I^2C and SPI communications, and features 55 I/O pins. The STM32F070RBT6 features twice the program and data memory at 128 kB and 20 kB respectively. It supports I^2C, SPI, and USART communications with 51 I/O pins. The chips feature nearly interchangeable pinouts as well, with the F070 variant sacrificing 4 I/O pins for 4 additional power supply pins. This 51 I/O pin arrangement is also pin-for-pin identical to the STM32F030RC, a variant of the F030R8T6 with 256 kB of program memory.

This line of chips boasts much higher performance than the MSP430 line at nearly the same cost. The F030 device is priced at around $2.11 per single-order chip while the F070 is around $4.70. STMicroelectronics also offers a line of development boards similar to the Launchpad from Texas Instruments called Nucleo. These boards feature full pin breakout in a form factor that is even compatible with Arduino shields. STMicroelectronics documentation and community support is also comparable to that of Texas Instruments, arguably even greater with respect to ARM devices. As a further bonus, all STMicroelectronics Nucleo Cortex M0 development boards come with a license for the professional version of Keil's MDK development environment for ARM. These points factor greatly when choosing a device as they all lend to an immediate boost in productivity during the early design and prototyping stages of the project.

3.4.4.4 STMicroelectronics STM32F103RBT6

Finally we examine the STM32F103RBT6. Like the previous devices, this offering from STMicroelectronics boasts an ARM Cortex series core, although here is the Cortex M3. The M3 core is similar in performance and specifications to the TM4C123GH6PM Cortex M4F offering from Texas Instruments but without the added DSP and floating point enhancements. It represents something of a compromise between that part and the STMicroelectronics Cortex M0 offerings.

At an impressive 72 MHz, the STM32F103RBT6 features the same memory characteristics as the STM32F070RBT6 with 128 kB program memory and 20 kB data RAM. As all of the Cortex M series parts examined it comes in a LQFP64 package utilizing a 32-bit data bus. It features 7 timers and support for I^2C, SPI, USART, CAN, and USB communications with 51 I/O pins. The price also represents a compromise between the M4F and M0 parts at $7.14 per chip when ordering individual pieces.

While the STM32F103RBT6 shares all of the same benefits of the M0 line from STMicroelectronics in terms of documentation and support, the increased performance may not be worth the added cost. The microcontroller in our project has relatively low processing requirements, and this M3 series part is likely similar overkill to the Texas Instruments Cortex M4F offering. While it lacks the DSP features of the TM4C123 and sells for a lower price as a result, it also lacks any particularly compelling features not found in the Cortex M0 alternatives other than a faster clock speed for the several dollars price premium.

3.4.4.5 Summary and Selection

The performance specifications most important to our project are compared by device in Table 6 below for convenience. Ultimately the STM32F070RBT86 best meets the needs of this project. Representing a cost and performance compromise relative to the Cortex M3 and M4F devices, this Cortex M0 chip boasts a significantly better set of specifications than the MSP430G2553 at a modest price difference. The addition of the STMicroelectronics community with the Nucleo development board and Keil MDK license makes this device an excellent all around solution.

Table 6: Comparison of Microcontroller options.

Device	MSP430	TM4C123	STM32F030	STM32F070	STM32F103
Clock	16 MHz	80 MHz	48 MHz	48 MHz	72 MHz
Bus Width	16 bits	32 bit	32 bit	32 bit	32 bit
Package	Various	LQFP64	LQFP64	LQFP64	LQFP64
Code Mem	16 kB	256 kB	64 kB	128 kB	128 kB
Data Mem	512 B	32 kB	8 kB	16 kB	20 kB
I/O Pins	Up to 24	Up to 43	55	51	51
Timers	2	12+	7	7	7
Price	$2.80	$11.00	$2.11	$4.70	$7.14

3.4.5 In-System Programming

As a main objective of the project is to implement all device components into a single circuit board, the programmable devices on board must support some form of in-system or in-circuit programming. These needs are different for the FPGA which must be programmed every time the device is reset and the microcontroller which can store its program binary in onboard flash. Proposed methods for programming of both devices are explored in this section.

3.4.5.1 FPGA JTAG Programming

Unlike microcontrollers which can store their programming file when power is lost, FPGAs require configuration upon each power cycle. This presents a challenge in transferring our design from the development board to the printed circuit board. We need a method to configure our FPGA to test our system and to update its configuration. JTAG (IEEE standard 1149.1) was originally introduced as a way to test electronic boards in order to trace and eliminate manufacturing issues. In our project we will be using JTAG to provide in-system programmability of the FPGA. JTAG is an easy to implement solution which requires a low number of pins and a small amount of board space.

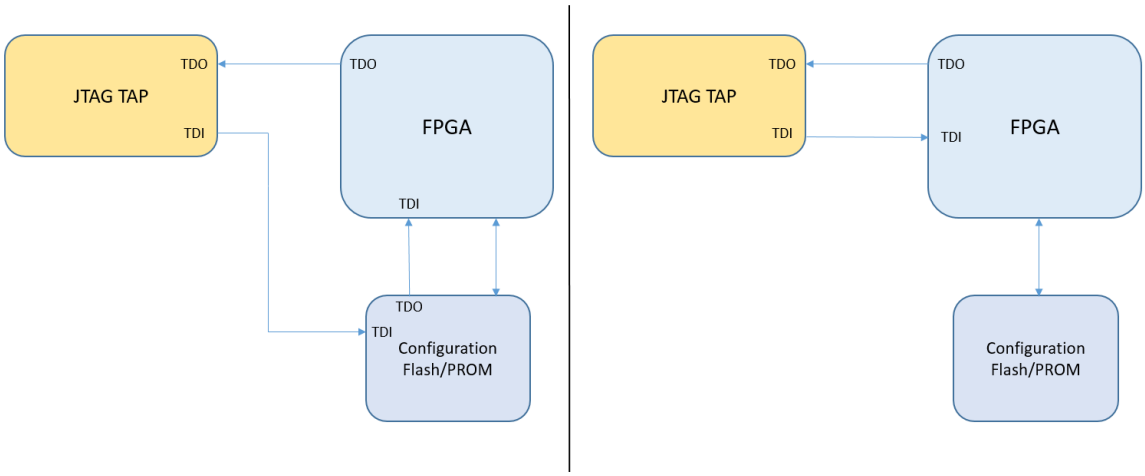


Figure 15: Direct vs indirect configuration.

Configuring of an FPGA using JTAG can be done in two main ways, directly and indirectly. The method used is determined by the designer's choice of configuration device and FPGA. With direct programming, when the configuration device/memory is on the scan chain, the configuration device/memory is programmed directly from the JTAG tap. When the FPGA is powered on, the configuration file is loaded from the external flash/PROM into the FPGA's internal SRAM. Indirect programming is used when the configuration device lacks a JTAG port and is not on the scan chain. In this case the configuration memory indirectly receives the configuration file, it is first passed from the JTAG tap into the FPGA and then into the configuration Flash/PROM. Just like with direct programming, the configuration file is loaded into the FPGA's SRAM upon power-on.

Direct programming of the configuration device represents the simpler of the two methods for our project. In our project we will directly program the configuration device since we will be frequently updating the FPGA throughout the design phase. Having a directly programmable configuration device will also make troubleshooting of our in-system programming easier since we do not have to confirm if a configuration file is correctly loaded into the FPGA, as is the case with indirect programming.

3.4.5.2 Microcontroller

Reprogramming and debugging the microcontroller unit in early prototype systems is extremely important. It is expected that some applications and subsystems in our project will not be able to be fully tested until a complete PCB is assembled. As the STM32F070RBT6 selected for use in the project is packaged as LQFP64, we must have the capability to program and debug without replacing the MCU IC to prevent significant risks from wear related to repeated soldering.

This is typically accomplished via a JTAG adapter. ARM specifies a reduced pin count alternative JTAG implementation called Serial Wire Debug. This debug interface at its

simplest requires just 2 pins, but the ST-LINK/V2 tool sold by STMicro utilizes the traditional 20 pin JTAG adapter. STMicroelectronics Cortex M microcontrollers also come with a firmware bootloader capable of loading programs via the chip's USART pins. This will be developed as the intended long term solution for simple program flashing and intended for use in the final prototype and by the end user if necessary.

3.4.5.2.1 Serial Wire Debug and ST-LINK/V2

Serial Wire Debug (SWD) is a 2 wire debug protocol developed by ARM for use with products utilizing ARM cores. The traditional 5 pin JTAG interface is replaced with a single clock wire (SWCLK) and a single input/output wire used for bidirectional communication (SWDIO in the STM32 documentation). This solution provides the resources needed for both reprogramming and interactive debugging of the ARM microcontroller.

STMicroelectronics sells a specific adapter intended to be used with their device's SWD capabilities. The ST-LINK/V2 is a JTAG/SWD programmer and debugger that connects to the developer's PC via USB and to the application board (for our purposes the project's PCB) via the standard 20 pin JTAG connector. The adapter requires just 4 active pin connections, specifically to the embedded microcontroller's SWDIO, SWCLK, power supply pin (V_{DD}), and reset pin (NRST). All other connector pins on the application board are connected to ground or unused.

3.4.5.2.2 STMicroelectronics Bootloader

STMicroelectronics Cortex M microcontrollers include a firmware bootloader capable of flashing the controller's program code space using the microcontroller's USART pins. To control the device's boot sequence, STM32 microcontrollers like the one selected for this project utilize two boot pins that are typically exposed with jumpers for configuration. As shown in Table 7, there are three bootable configurations for the microcontroller.

Table 7: STM32F070RBT6 boot modes.

nBOOT1	BOOT0	Boot mode (source used)
x	0	Main Flash memory
1	1	System memory (bootloader)
0	1	Embedded SRAM

Also in the reference manual for the STM32F070RBT6 it is specified that the memory location specifying the bit nBOOT1 defaults to 1 on device reset. This means by jumping only BOOT0, which is exposed as a pin on the microcontroller package, to either 3.3V or Ground we can toggle between running the currently loaded code or the factory bootloader for flashing a new binary. STMicro also provides a program for this purpose, the Flash Loader Demonstrator, which is capable of programming the device over any interface exposing the USART pins.

3.4.6 USB Interface

While the full SWD JTAG-like interface provides a fully featured developer interface to programming and debugging the microcontroller, the connections and tools necessary for its use are not ideal for software updates to be deployed by an end user. To this end, a USB interface for reprogramming is desirable. This interface should support flashing the microcontroller with a complete program binary using a flash programming program and is intended to be the programming interface used by all users except project developers.

3.4.6.1 Bootloader

The key to support for this feature lies with support for a separate bootloader program for flashing the program memory using a binary delivered over the USB interface. The bootloader program factory loaded onto modern STM32 Cortex-M microcontrollers provides for such a feature rather directly. At power on, configuration of the BOOT0 pin as HIGH will cause the bootloader code to be loaded into memory rather than the user flashed binary. This bootloader program is capable of reprogramming the STM32 microcontroller using the device's USART1 UART-capable channel. This solution does however necessitate that the BOOT0 pin is exposed via a jumper to allow the user to easily select to reprogram via this interface.

3.4.6.2 FTDI FT232

The Future Technology FT232 is one of the most ubiquitous USB to serial UART chips in the world. It is responsible for conversion between USB serial data streams and any of the various UART protocols (RS232, RS422, RS485, etc). As traditional 9-pin serial interfaces are phased out from most computers, the FT232 allows use of the surely available USB ports to access functionality provided over a target device's UART pins. Data received over the USB Data+ and Data- pins are converted and corresponding data is sent over the FT232's Tx pin to the microcontroller (and the opposite FT232 Rx to USB path).

The FT232 supports power supply input between 3.3 and 5.25 V, and provides a separate pin for determining the IO pin voltage levels to support logic levels used by a wide variety of targets between 1.8 and 5.25 V. These configurations also support direct powering from the typically 5 V USB bus power line. Provided the MCU is also powered from USB when connected, this could allow reprogramming to be supported using only the USB connection with no external power supply necessary.

The FT232R costs approximately $4.50 per unit. It is available in both a QFN32 and SSOP28 package. The latter package (FT232RL) is found in the Sparkfun FT232RL Breakout board for approximately $15, shown in Figure 16. The group has one such breakout board on hand from previous projects and the board can be used for early testing of the USB interface feature. Additionally, the board could be implemented into the project

as a separate broken out PCB to reduce the footprint of the main device PCB and reduce project costs.

Figure 16: FT232RL FT232RL breakout board. Courtesy of Sparkfun.

3.4.7 SD Cards

While the USB interface can be used to reprogram the final product using the STM32 microcontroller's pre-installed bootloader (as well as the ST-LINK/V2 debug port), the project also requires an easy way of reprogramming the FPGA logic. Secure Digital (SD) Cards are a solution to this problem, allowing the user to use a user-generated or supplied .bit file for FPGA initialization or changing of the stored FPGA binary.

SD cards are the standard non-volatile memory card used in applications requiring a small footprint, high-capacity removable storage solution. This is the primary reason for their selection for FPGA reprogramming interface in the project as they are readily available in many form factors, sizes, and for an extremely low cost. Potential users of our project likely already have several and so incur no extra cost to support the device. As of this writing 4 GB SD Cards, far larger than necessary for supporting the project's reprogramming features, can be readily found for under $5.

As mentioned, SD cards are available in several form factors. While microSD has become extremely popular in recent years, the main SDC form factor remains the standard. Adapters are available for converting the pinouts of other form factors to the SDC pinout. For this reason, we have elected to focus on supporting the SDC interface with a single SDC jack.

The SD specification defines three transfer modes: use of the SPI bus, a one-bit bus mode, and a four-bit bus mode. As our microcontroller does not have a dedicated SD interface we must either dedicate the GPIO pins necessary to control using the SD bus modes or use one of the microcontroller's built in SPI busses. SD cards have a modified pinout when used in SPI mode. These pins can be used to interface to the microcontroller SPI pins for control of and data transfers to/from the SD card.

The final consideration with the SD card is that SD cards themselves are block storage devices typically formatted using some filesystem, usually FAT16/FAT32. Therefore we must have a means of manipulating this filesystem for transferring data rather than accessing specific memory locations directly. There are several practical implementations available compatible with our expected software ecosystem including an open source FAT driver library called FatFs and a commercial middleware solution for filesystem access available through Keil (that unfortunately requires a professional edition of Keil MDK).

3.4.8 Supported Connectors

There are several good options when it comes to the connectors available to coupled the input and output from the peripheral entertainment unit and the digital television. A primary requirement of our project is to support a wide array of input connectors to support the most legacy devices as is possible. Among the options available are SCART, component, VGA, HDMI and stereo connectors. Discussed below is the pros and cons of each connector included in the design of our prototype.

3.4.8.1 SCART

SCART is a 21-pin connector, created by the French that was the standard connector utilized in Europe to connect audiovisual equipment. SCART allows analog video and audio to be sent within one cable however is becoming more obsolete as digital video becomes the new standard. Despite the fact that it is not utilized with newer technology, it is still frequently used and can even support HD signals from 480p to 1080p if connected to an RGB video supported device. SCART can support composite, RGB and S-video signals coupled with stereo audio and digital signaling. The unique aspect of the connector is its 21 pins that allow the cable to send video and audio signals in two directions as well as AV Auto Switching, Wide Screen Switching and RGB Status Switching. This type of connection is beneficial in that it allows an entertainment peripheral to mandate a TV to switch quickly between different signals for a clearer image rather then having to go back and send a completely new video signal that needs to be encoded and decoded. A fully wired SCART connector refers to a cable that has the ability to carry additional AV signals with audio such as composite, YB, and RGB. One of the best aspects of the SCART cable that speaks to its versatility is that you can connect the SCART with other connectors. Some TV sets have two SCART connectors where the first cable can switch from composite input to RGB input and the second one can switch from composite input to S-video input. Also available is the SCART switch that allows you to connect multiple devices to your TV if your TV only has 1 SCART component. The switch lets the user switch electronically between different devices such as DVD players and video game consoles.

The benefits of utilizing the SCART connections is that it can easily be coupled with legacy video game consoles such as SNES or Sega due to those systems automatically support RGB signals and do not have to be manually switched from composite to RBG, as with newer technology. Digital devices often default to composite video which often distorts the

image if not manually switched to RGB output. In addition, older video game units support SCART which can easily be coupled with an HMDI connector for improved video output quality. One drawback of the SCART connection is that is does not process HDMI unless the component is compatible with YpBPr. Below is a table depicting the SCART 21 pin configuration and SCART interface.

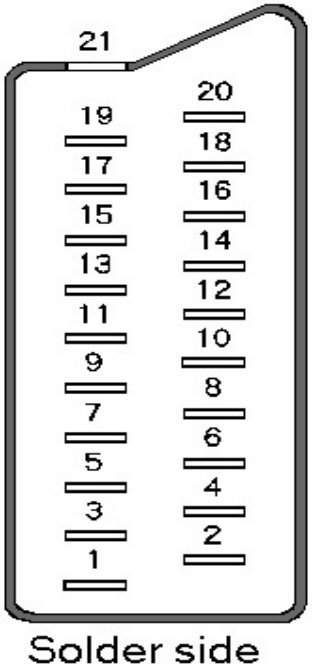


Figure 17: SCART socket pinout. Penguin.com permission pending.

Table 8: SCART pin layout.

Pin	Signal	Level	Impedance
1	Audio Out Right	0.5 V rms	<1k ohm
2	Audio In Right	0.5 V rms	>10k ohm
3	Audio Out Left + Mono	0.5 V rms	<1k ohm
4	Ground Audio		
5	RGB Ground Blue		
6	Audio In Left + Mono	0.5 V rms	<10k ohm
7	RGB Blue In	0.7 V	75 ohm
8	Audio/RGB switch / 16:9	High (9.5-12V) AVmode Low (0-2V) TVmode	<10kohm
9	RGB Ground Green		
10	Comms Data 2		
11	RGB Green In	0.7 V	75 ohm
12	Comms Data 1		
13	RGB Ground Red		
14	Ground Data		
15	RGB Red In / Chrominance	0.7 V (Chrom.: 0.3 V burst)	75 ohm
16	Blanking Signal	High (1-3V) RGB Low (0-0.4V) Composite	75 ohm
17	Ground Composite Video		
18	Ground Blanking Signal		
19	Composite Video Out	1V including sync	75 ohm
20	Composite Video In	1V including sync	75 ohm
21	Ground/Shield (Chassis)		

3.4.8.2 Component

Component video connectors, also known as YPbPr, is an AV cable that transmits analog video through 5 connectors comprised of red, blue and green cables for video and a left and right audio cable. The red cable (Y) transmits black and white video information. The blue and green cables transmit color information. Benefits of using the component video connector is that though designed for analog video it can be used with most modern televisions to connect DVD and other peripherals. This type of connector sends the analog video data in a stream of varying voltages that represents the red, blue and green elements of the signal. The different color elements along with sync information (the Y cable) travels through the connector to allow the display to decipher when a new frame begins. The component connector is very similar in color definition and resolution as the HDMI connector with the difference being that the component connector utilizes analog while HDMI works for digital video. The component cable also allows for progressive scanning (480p) where all lines of a frame are drawn in sequential order one at a time. One of the drawbacks to the component connector when running analog video or when converting analog to digital through the component connector is that analog signals are susceptible to deterioration and loss of data. Often RCA plugs are utilized for composite video.

Figure 18: A set of RCA component video jacks. Courtesy of bluejeanscable.com.

3.4.8.3 VGA

VGA stands for video graphics array and is a 3-row connector with 15 pin slots that carries analog RGBHV signal. This type of connector, also referred to as SVGA, is found most often on computers and HDTV and can come in both mini and full size. Though commonly found on many computers and monitors, it is slowly being replaced with DVI connector and cable. The cable associated with the VGA connector is vital in determining the resolution of the video since different cables support different bandwidth and the higher quality cables contain coaxial wiring with insulation. One of the drawbacks to this type of connector is that within the 15 pins, the red, blue and green signals (pin 1,2, and 3) are not separated and crosstalk between the signals is possible. The standards for this type of connector are 640x480 resolution with a refresh rate of 60 Hz.

Table 9: Specifications for VGA.

VGA	
Specifications	RGB video with 15 pins
Cables	Analog only
Compatibility	Can convert VGA to DVI and VGA to HDMI
Image	Maximum resolution 2053x1536 but output image is distorted due to issues with signal conversion from analog to digital and the crosstalk between RGB

3.4.8.4 HDMI

Given that our project's goal is to provide a way to interface analog video signals with digital televisions, it makes sense to use HDMI as the primary video output. A benefit of using HDMI is that we can use a single cable to carry both audio and video signals. Using a single cable provides a reduction in board space for the audio and video connectors and simplifies device setup for the consumer. The robust TMDS link used in HDMI also allows for very long connecting cables from device to television without risk of signal degradation.

The various HDMI connector types are shown in Figure 19 below. Type B is not shown due to it being far too large for our project and type E is not shown as it is typically used only for automotive applications. All connectors featured below support the HDMI standard in full, the 19 pin arrangement for the Type A HDMI connector is given in Table 10. While Type C and Type D HDMI connectors incur the lowest board space cost, their use is not as widespread as the Type A HDMI connector. As a result cables with Type A connectors on both end are more common; Type A cables are also available in longer cable lengths. We chose to support the Type A HDMI connector so that the user doesn't have to track down obscure variants which may require repeaters to achieve the desired routable cable length.

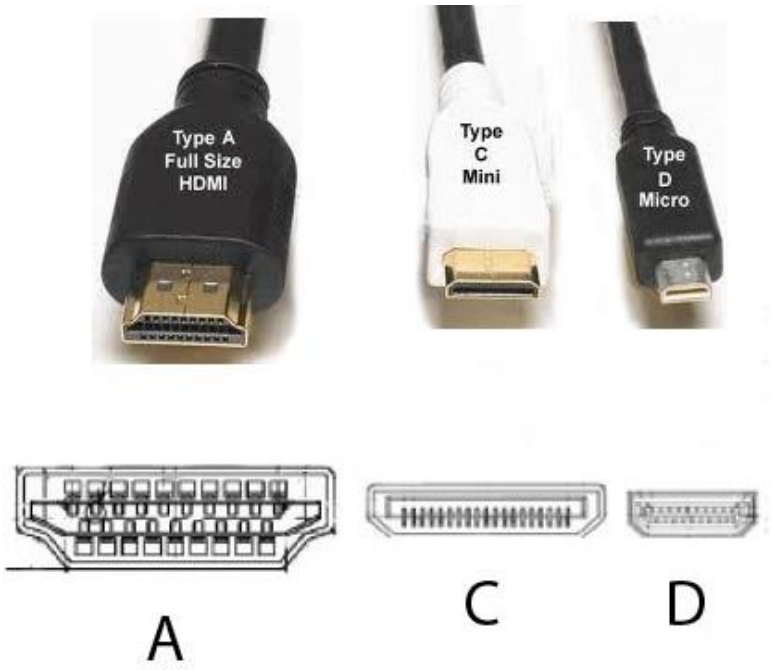


Figure 19: HDMI A, C, D connectors. Courtesy of Jan deno via CC BY-SA 3.0.

Table 10: Comparison of HDMI connector dimensions.

Dimensions	
Type A	14 mm × 4.55 mm
Type C	10.42 mm × 2.42 mm
Type D	6.4 mm × 2.8 mm

The pinout for a Type A HDMI connector is given in Figure 20 below, a brief description of each pin is given in Table 11 below. All HDMI sources must be able to supply +5V at 0.055A when using DDC or TMDS signals. Our device will not support the latest HDMI standard and certain features like Pin 19's Ethernet channel will go unused.

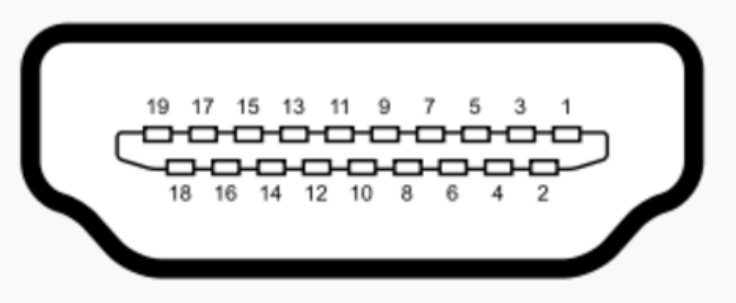


Figure 20: HDMI type A connector pinout.

Table 11: HDMI Type A pinout description.

Pin 1	TMDS Data2+	**Pin 11**	TMDS Clock Shield
Pin 2	TMDS Data2 Shield	**Pin 12**	TMDS Clock−
Pin 3	TMDS Data2−	**Pin 13**	CEC
Pin 4	TMDS Data1+	**Pin 14**	Reserved (HDMI 1.0–1.3c), Utility/HEC/ARC
Pin 5	TMDS Data1 Shield	**Pin 15**	SCL (I²C Serial Clock for DDC)
Pin 6	TMDS Data1−	**Pin 16**	SDA (I²C Serial Data Line for DDC)
Pin 7	TMDS Data0+	**Pin 17**	DDC/CEC/ARC/HEC Ground
Pin 8	TMDS Data0 Shield	**Pin 18**	+5 V (min. 0.055 A)[3]
Pin 9	TMDS Data0−	**Pin 19**	Hot Plug detect (all versions)
Pin 10	TMDS Clock+		

3.4.8.5 Stereo Audio

Stereo Audio sounds consists of utilizing two different audio signals, usually a left and right, and mixing them together to create the effect of sound coming from more than one direction. Compared to mono audio (single channel), stereo usually produces a more realistic sound since it mimics human hearing more closely. Mono audio is only beneficial over stereo during speeches due to the fact that everyone would hear the same signal and at the same sound level so it allows for improved intelligibility during speeches. Since stereo audio incorporates two separate signals for the left and right, it is important that both signals have to have equal scope of the listening area or some listeners may only hear half of the sound. There are several different options for analog audio connectors including RCA connectors and SCART connectors.

The RCA connectors (also known as A/V jacks) utilized for audio have a female jack and the cable consists of a male end and a standard plug. These connectors can be used with composite, component and even digital audio when coupled with S/PDIF. A disadvantage of the RCA is that it can give off a buzz sound when in use. In addition, when using a RCA connector for audio, a different cable is needed for each signal.

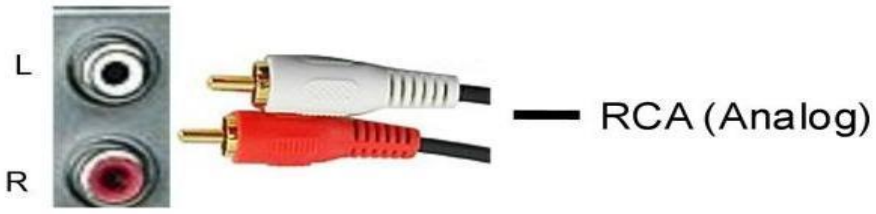


Figure 21: RCA Audio connector. Courtesy of thehdstandard.com.

The SCART connector handles bidirectional stereo, which like the HDMI connector for digital audio, is able to reroute audio from the television to an external source such as a home surround sound or amplifiers when coupled with a SCART adapter. Many retro gaming systems are able to utilize the SCART RGB setup.

Figure 22: SCART to RGB audio connector. Cablemagic.au. permission pending.

3.4.9 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 and usable anywhere in the world. Consequently, we choose a video decoder that is compatible with all major consumer video input formats. In our project we are using the ADV7181, which is compatible with NTSC, PAL and SECAM analog signal. This particular decoder allows for multiple different signal input formats such as composite, S-video, and the component standard YPrPb. Additionally, the chip is programmable allowing significant modification and customization of the resulting output. In addition, this decoder allows for 6 different analog input channels allowing simplification of the input network; each of the analog inputs can be directly connected to one of the input channels on the video decoder chip.

The digital output network on this decoder utilizes in line-locked clock (LLC) based systems and allows for signal peak-to-peak range of .5 V to 1.6 V, which the user can override manually. Basically the video decoder automatically detects and converts standard

analog baseband television signals. Most video decoders can be programmed to allow the viewer to adjust the contrast and color saturation of the video.

The pin configuration, or pin out, of the decoder chip is shown below in Figure 11. As illustrated, below pins 35, 36, and 46-49 relate to the analog video input channels. Pins 23, 58, 40, and 31 incorporate the analog supply voltage, digital supply voltage and the PLL (phase locked loop) supply voltage. Synchronization output is located at pins 2, 64, and 63 and the video pixel out is providing on P0-P15. Lastly pins 52, 53, and 54 encompass the I2C data output for the decoder to the MCU.

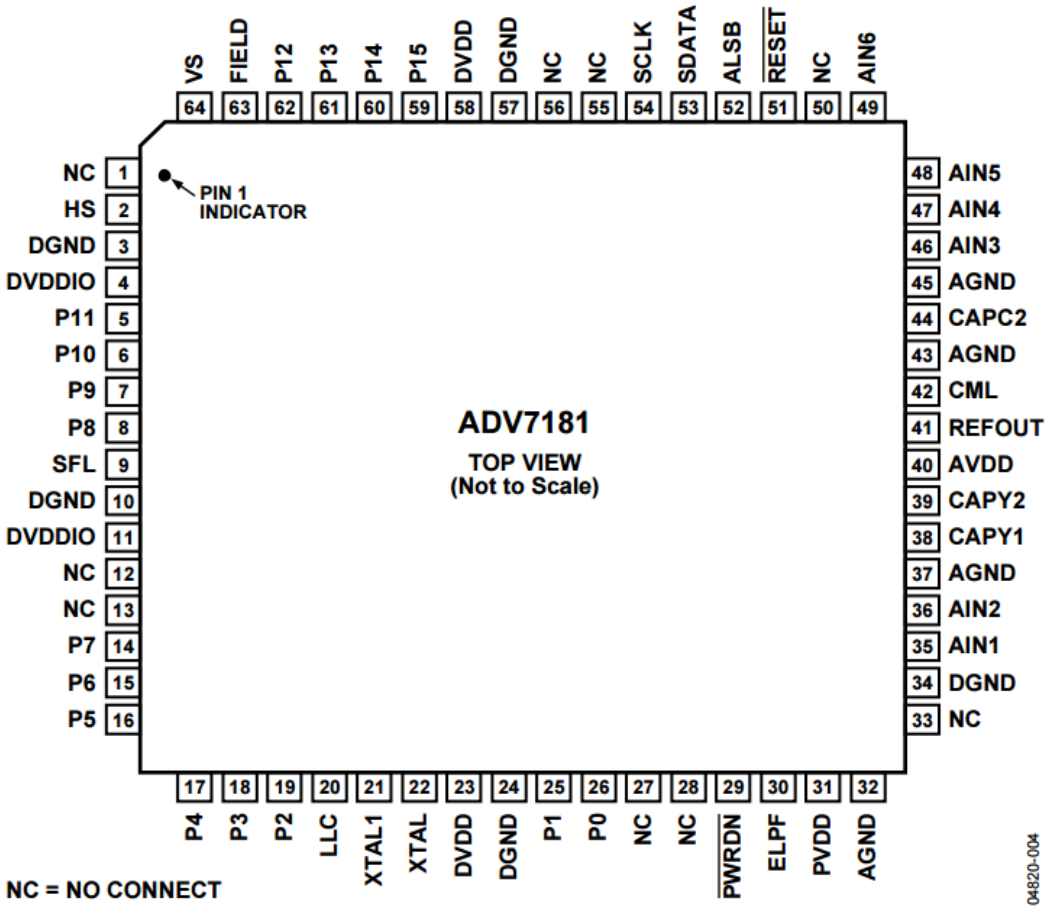


Figure 23: Pin Configuration for ADV7181. Courtesy of Analog Devices.

3.4.10 Audio Decoder

In addition to processing video from the input device, we also must process analog audio. This audio may be mono or stereo and could vary in quality. To capture this audio we use an analog to digital converter, or ADC. We will be using the AD1871 because it utilizes 24-bit conversion channels. Decoders with 24 bit conversion provide vastly superior sound quality and much greater dynamic range. The AD1871 also provides a PGA (programmable gain amplifier) which allows the input level to be put under CPU control,

which may be necessary for differing inputs. Each of the two individual channels of the stereo AD1871 allow for 105-db range of sound.

This analog to digital converter, or ADC, supports audio sampling rates from 32 kHz to 96 kHz. The ability to support rates up to 96kHz provides higher quality sound. The audio data interface supports and connects to a SPI (serial peripheral interface) bus which is a high speed serial bus that supports the necessary high data rates and allows for manipulation of the PGA settings and interface modes. As shown in Figure 12, pins 2-5 relate to the SPI interface, pins 6 and 9 are the analog and digital power supply, pins 10, 11, 18, 19 are the left and right input and pin 26 is audio output.

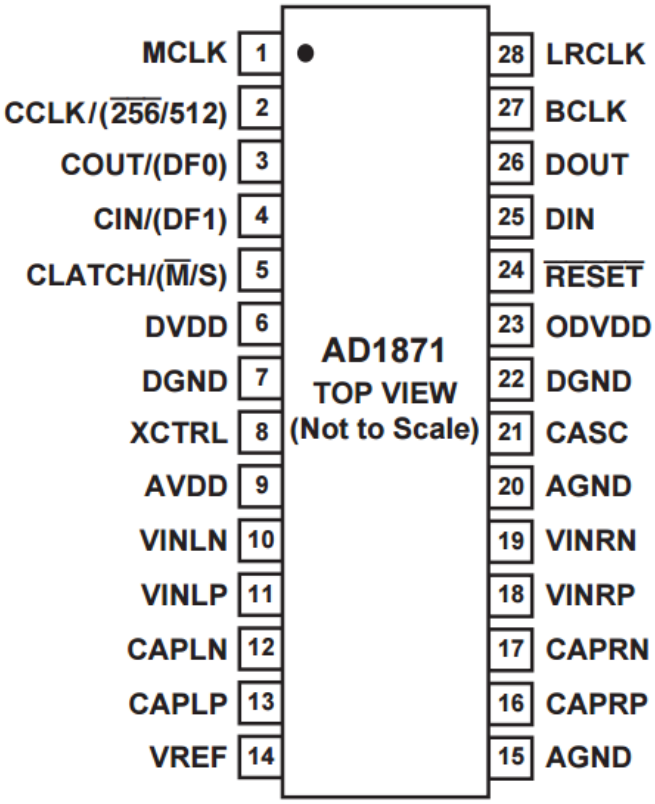


Figure 24: Pin configuration for the AD1871. Courtesy of Analog Devices

3.4.11 HDMI Transmitter/Encoder

While the FPGA is well-equipped to handle image scaling and basic control tasks, the encoding of a digital video signal such as HDMI can be very difficult. In addition to the FPGA hardware design challenges, there are also physical issues such as generating the required differential signal directly from the FPGA output. The use of an HDMI transmitter integrated circuit simplifies system design and provides reliable HDMI encoding which will be compatible with most displays. The specific HDMI transmitter chosen for our project is Analog Device's ADV7511 225MHz HDMI transmitter. The supported pixel

clock of 225MHz is well below the HDMI upper limit of 340MHz but this shouldn't be an issue given that our target pixel clock is 25MHz-50MHz.

Figure 25 presents a high-level view of what the transmitter is accomplishing and overall the purpose for using the HDMI signal in our design. Without such a transmitter we would not be able to reasonably transmit the data to a digital display. The ADV7511 also supports the encoding of a digital audio signal across the TMDS channels which simplifies audio integration within the output signal. The ADV7511 allows us to convert three color channels of eight bits each, totaling 24-bits, into three differential signal pairs. Not only is the wire count reduced, but signal integrity is improved through the use of differential signaling and twisted pair wiring. The ADV7511 operates at a supply voltage of 5V and has a maximum operating temperature of 70°C. The chip is available in a 100-lead LQFP surface-mount plastic package at a reasonable cost of $14.88.

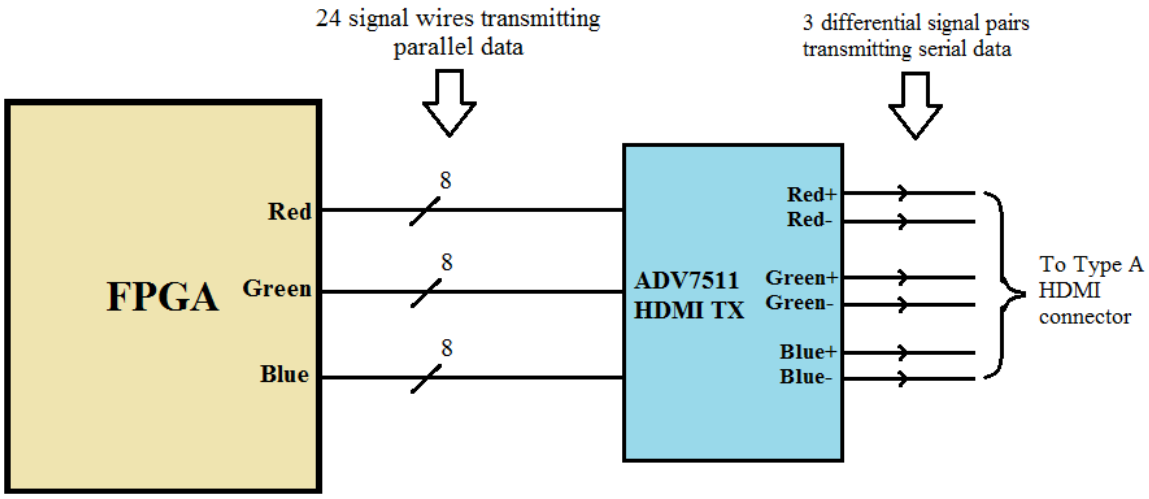


Figure 25: Add extra wire called "timing" to represent all sync info.

3.4.12 VGA Output Support

In addition to supporting HDMI output, our device will be able to output video using a video graphics array (VGA) cable. The reason for including VGA output is to have a line doubling device which can also be used on CRT displays which support 480p. While that may seem counterintuitive to the initial goal of the project, VGA output is a feature commonly requested in video scaling devices. A VGA output further increases the flexibility and compatibility with displays.

A VGA video port sends the video using five signal wires; the analog signals are red, green, blue, horizontal synchronization, and vertical synchronization. The red, green, and blue signals will be passed from the input decoder to the FPGA where the line doubling will occur. After the line doubling the signals will be sent to the FPGA's output pins as 24-bit RGB. The synchronization signals will be briefly passed through the FPGA to take

advantage of the clock management tiles which can align the synchronization signals with the 24- bit RGB output signals.

Since the FPGA outputs only digital signals, some form of digital to analog conversion is required in order to obtain a VGA signal. Conversion solutions like binary-weighted DACs and pulse-width modulation were disregarded and a pure resistor ladder scheme was chosen due to its simplicity. The R-2R schematic in Figure 26 is what will be used for each set of color signals. A buffer stage is required at the analog output node to prevent impedance matching issues at the output. While this solution is expensive in terms of FPGA pin count, it is cheap in terms of parts cost and design time. The Artix-7 XC7A35T has 170 I/O pins in the 256-LBGA package. The R-2R DAC requires a total of 24 output pins and the VGA output in total requires 26 pins from the FPGA. It may also be possible to share the 24-bit RGB output lines being used as input for the HDMI transmitter.

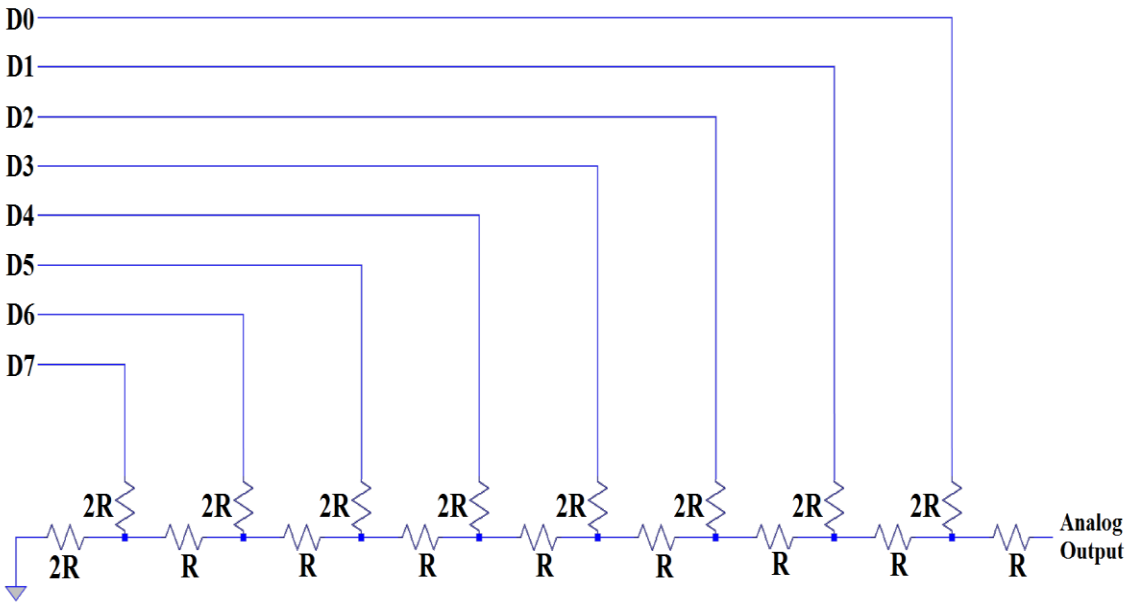


Figure 26: Schematic of 8-bit R2R DAC for VGA output.

3.4.13 LEDs

LEDs will be utilized on the main project enclosure as a means of providing immediate visual feedback to the user concerning the device's state of operation. Detected power supply, internal device communications verification (for example successful communications between MCU and FPGA), input signal detection status, and output device connection are among the uses of LED feedback for the device. LEDs in a variety of specifications and packages are available to support this feature. Through-hole solderable discrete single-color LEDs should satisfy expected physical requirements such as size, brightness and color support and provide the lowest cost and lowest PCB footprint

with easy installation. Typical 3 mm or 5 mm T-1 packaged devices meet all of the above needs.

The most important design considerations for the LEDs are forward voltage drop and current draw necessary for adequate brightness. As there are few LEDs used by the device it is desirable for simplest implementation to drive them directly via the microcontroller's I/O pins with appropriate current limiting resistors. Microcontrollers in this device class typically have relatively low per pin and total current source/sink limitations, often around 20 mA per pin and 100-200 mA total source/sink limits. The microcontroller may also not be 5V output tolerant, and so the LED model selected should have a forward voltage drop of less than the expected 3.3V able to be supplied by the microcontroller's I/O pins.

LEDs are easily acquired from an enormous number of sources. As such, the most efficient means of choosing the appropriate pieces is by comparison of typical LED characteristics by color and experimentation. Typical LED characteristics for single-color discrete LEDs from a sample vendor search and analysis for the colors under consideration are shown in the table (TABLE NUMBER) below. LEDs examined were tested and evaluated typically at 20 mA supply current. Appropriate operating characteristics will need to be determined experimentally for given LED brightness observations and current requirements.

Table 12: Comparison of LED characteristics by color.

Color	Approximate Fwd ΔV
Red	1.6 – 2.2 V
Green	1.8 – 3.4 V
Yellow	1.8 – 2.4 V
Orange	1.9 – 2.4 V
Blue	2.6 – 4.0 V

Surveying of multiple distributors yields ready availability of the parts in question in packages evaluated for standard brightness at currents of between 1 and 5 mA, much more desirable than the typical 16-20 mA operation for use in our project. Though these must be purchased individually at a higher price per piece (approximately $0.50/piece compared to bulk costs as low as pennies each) than typical bulk acquisition of LEDs, they provide a valuable alternative if no bulk or variety-pack solution is found to meet the project's needs.

3.4.14 Remote Control

The Super-Doubler will allow the user to select inputs and set scaling options remotely via a remote control. The remote control will send either a Bluetooth or infrared signal to a receiver on the unit and will be decoded by the microcontroller which will make the changes as directed by the user.

There are several differences between the Bluetooth and infrared technology that must be considered when deciding on the method of remote control communication for this particular project. The remote control will only give a small number of commands to the

Super-Doubler so a complex system is unnecessary for this task. Also, the remote control will not need to connect to any device other than the Super-Doubler. Since this is the case, the system can be designed to receive signal from one type of device. Taking the simple requirements of this system into account, cost will play the biggest role in determining the technology used for the remote control system of this device.

3.4.14.1 Bluetooth

Another form of wireless communication to be considered for the remote control system is Bluetooth. Bluetooth is a wireless communication standard used to connect devices over short distances. Bluetooth is used in a very wide variety of products including but not limited to smart phones, desktops and laptops, wireless keyboards and mouses for computers, wireless speakers, car stereos and hundreds of others. It certainly has become the global standard for short range communication between electronic devices and is the means of communication for the Internet of Things. While infrared transmitters and receivers operate in the infrared spectrum (300 GHz to 430 THz) Bluetooth technology transmits signals using ultra high frequency radio waves from 2.402 GHz to 2.48 GHz with a guard band of 2 MHz at the low side and 3.5 MHz at the top side: the overall bandwidth ranges from 2.4 GHz to 2.4835.

Bluetooth utilizes frequency-hopping spread spectrum when transmitting and receiving data from other Bluetooth devices. Frequency-hoping spread spectrum involves transmitting packets of information over channels that subdivide the effective bandwidth. The individual channel bandwidth utilized in Bluetooth technology is 1 MHz wide therefore employing the use of 79 channels. The technology will perform around 1600 “hops” per second to transmit and recover the data. This technology utilizes the entire prescribed bandwidth when devices are communicating.

There are several different types of Bluetooth available for designers to utilize. One of the most common forms of Bluetooth is known as BR/EDR (Basic Rate/Enhanced data Rate). Basic Rate Bluetooth has a data rate of 1 Mb/s while the Enhanced data rate supports up to 2 Mb/s. This form of Bluetooth, also known as Bluetooth Core Specification Version 2.1, allows devices to automatically find and connect to one another once selected by the user. Bluetooth Low Energy, also known as Bluetooth smart, is a power efficient version of Bluetooth that was originally designed for use in the internet of things. Bluetooth Low Energy is ideal to use in devices with limited power sources such as batteries. Bluetooth High Speed allows for the quick sharing of music, video and other large media files. The use of a second (high speed) radio transmitter is used which increases the data rate of the of communication. This high speed radio turns on only momentarily to help conserve energy.

Bluetooth technology is not a line-of-sight based form of communication meaning the transmitter does not need to be pointed directly at the receiver in order for it to receive the signal. Bluetooth devices can often times have physical obstacles such as walls between them and still communicate effectively. Up to 8 devices can be connected together at once; one device as the master and the other seven as slaved devices. This allows for

interoperability with supported devices. There are several resources available to designers who choose to use this technology. Bluetooth Special Interest Group has created many resources in the forms of webinars, training videos and datasheets to assist those unfamiliar with Bluetooth in design and application. There are several consumer products on the market that allow the utilization of Bluetooth communication. However, Bluetooth technology may be too complex for what is needed in the Super-Doubler's remote control system.

3.4.14.2 Infrared

Infrared remote control is the most popular form of remote control that will be considered for this project. Most house hold appliances that utilize remote control do so using infrared transmitting and receiving (televisions, garage door openers, wireless computer mice and key boards, etc). Infrared remote control is an excellent form of wireless communication because it utilizes extravisible light to transmit an electrical signal through air. Since the light is undetectable to the human eye, infrared communication poses no health risks (unlike lasers).

Infrared transmitters use an LED (IRLED) to transmit the light waves through the air. When the user presses a button on the remote control, the IRLED pulses the code associated with that particular button. To preserve the life of the IRLED, the pulse code is modulated thus reducing the amount of power consumed by the transmitter. Typical modulation carriers are between 30 kHz and 54 kHz. The signal is received, demodulated and passed to a microcontroller for further processing. The microcontroller receives this series of pulses and reads them as logic "1's" and "0's".

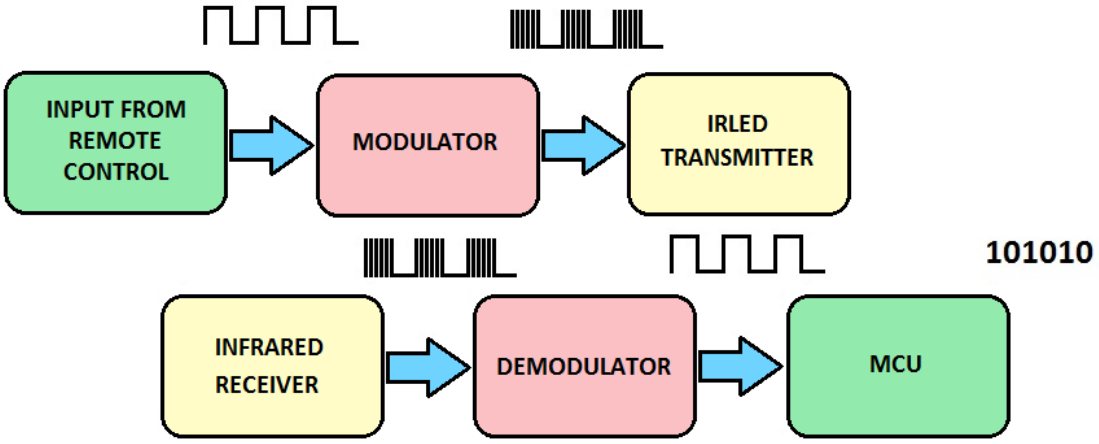


Figure 27: Infrared signal path.

The world or infrared remote control is practically unstandardized and companies that utilize this technology have created their own pules codes to allow their devices to communicate with each other. Transmitters will send a series of pulses (dictated by the chosen communication protocol) to let the receiver know a signal is about to be transmitted. If the receiver is not programmed to receive signals of that particular protocol the signal

will not be processed and the desired action will not take place. When implementing infrared remote control, the receiver must be set up to receive the signal according to the protocol used by the transmitter.

IRLED's typically have a wide range of transmission which compensates for some use outside of the line of sight of the receiver. Infrared technology is a line-of-sight based form of communication that requires the transmitter to be "in sight" of the receiver, though many modern infrared receivers have very wide viewing angles allowing for some degree of freedom. Infrared transmitters and receivers are also inexpensive and relatively easy to work with. Modern infrared receivers receive, demodulate and pre-amplify the signal in the same package and outputs a signal ready to be used by a microcontroller or other processor.

When choosing the infrared transmitter and receiver for this project, several parameters must be considered. First, the amount of functions the remote control will allow the user to perform must be determined. Purchasing a remote control with the minimum amount of functions is ideal in terms of cost, ease of design and implementation, and ease of use. Secondly, the communication protocol must be considered. Several protocols exist and many allow for the interoperability of compatible remote transmitters. Next, transmitting and receiving ranges and receiving angles must be accounted for. Modern IR receivers have wide receiving angles but infrared technology does not allow for the transmission of signal through mediums other than air. This system will operate in a small range (same room) but the receiver must be able to receive the signal in a wide angle window.

REMOTE CONTROL

An infrared remote control will be used to give commands to the MCU to perform tasks such as selecting inputs, adjusting color brightness, select scaling options and to activate scan-line emulation. Because the IR remote control will be operated in close proximity to the Super-Doubler unit (and IR receiver), cost will be the most important factor in determining the IR remote control that will be used. The remote control should have the minimum inputs required to perform the listed tasks and to not cause the user any confusion; having buttons on the remote that have no assigned function. sparkfun.com offers an infrared remote control that fits this requirement. The remote control has four directional buttons, power button, and three select buttons. The remote control unit requires a battery to be purchased separately. The cost of the remote control is $4.95 per unit and the cost of the battery is $1.95 per unit summing to a total cost of $6.90. Both parts are available to for online purchase at sparkfun.com.

The same infrared remote control and battery can also be purchased in a kit along with 2 infrared receivers, 2 IR LED's and 25 current-limiting resistors on sparkfun.com. The IR receivers that come in this kit are through-hole parts ready to be mounted to the PCB. Even though it is desired to use surface mount components, this kit would provide IR receivers that can be easily implemented in a breadboard circuit to test and troubleshoot the IR remote control system. Surface mount receivers will be used for the final project. This kit can be purchased at sparkfun.com for $9.95 and are readily available.

The IR remote control kit provides the ability to create test circuits for the IR remote control system as well as the remote control that will be used in the final project. The remote control is very simple while allowing the designer to have several input options for system control. sparkfun.com provides several references to help the designer implement the system. Test code is provided for use with Arduino MCU's and several video tutorials are available.

VISHAY SEMICONDUCTORS TSOP853..

Vishay Semiconductors manufactures a family of infrared receiver modules with some notable and attractive features. This family of IR receivers comes with the photodetector and preamplifier in the same package. This receiver contains two dome-shaped infrared filtering lenses that contribute to the receiver's strong immunity against ambient light and allow for a wide viewing angle of -50 to +50 degrees and impressive 45 meter sensing distance. The TSOP853.. family of parts receives signals modulated on many different carrier frequencies making these receivers a strong competitor for any IR RC system. The supply voltage needed for this receiver is 2.5 volts to 5.5 volts and is insensitive to supply voltage ripple. TSOP853.. family receivers are surface mount parts that are capable of side view and top view at and are very small (mounting surface are is 3.3mm x 8mm) preserving area on the printed circuit board. The TSOP853.. family is an attractive option when considering IR receivers for this project.

VISHAY SEMICONDUCTORS TSOP6...

The TSOP62.. and TSOP64.. are two other families of IR receivers manufactured by Vishay Semiconductors. This particular receiver is optimized to suppress almost all pulses of IR interference and has improved shielding against electro-magnetic interference. Like the family of IR receivers previously mentioned, these receivers demodulate the signal at several different carrier frequencies and output a ready-to-use signal to be decoded by the microcontroller. The typical viewing angle is -50 to +50 degrees and can sense signals from a range of 40 meters. This receiver is also insensitive to supply voltage ripple and requires 2.5 volts to 5.5 volts to operate. This part is available in either a top-view receiving package or a side view receiving package.

VISHAY SEMICONDUCTORS TSOP75...W

These two families of IR receivers are also manufactured by Vishay Semiconductors and contain many of the same characteristics as the IR receivers previously discussed. This particular family contains the photodetector, demodulator and preamplifier all in one low profile package. This IR receiver has an infrared filtering lens that contributes to its advertised immunity against heat emitting sources. The optical filter is narrow to reduce interference from plasma TV emission which would benefit this project since the device will be used around televisions (potentially plasma TV's). The one characteristic that sets this receiver apart from the others is that it has a very wide receiving angle or -75 to +75 degrees and can sense signals from a range of 30 meters.

COMPARISON

Since all of these parts are manufactured by the same company and provide the same basic service, all families of parts contain attractive features and characteristics as Infrared receivers. There are a few characteristics, however, that set them apart from each other. Sensing angle and distance are amongst the deciding factors. The TSOP853.. and TSOP62.. families of IR receivers have the same viewing angles and similar sensing ranges which would suffice for this project. The TSOP75…W has the shortest sensing range but the widest viewing angle. Sense this final product will most likely not be operated from more than 10 meters away, the sensing ranges of these receivers is not important. TSOP853.. and TSOP75…W allow signals to be received from either the top view or the side view of the receiver while the TSOP6… family offers parts that can sense signals from the top view or side view, but not both. Cost and availability are other very important parameters since most of the characteristics of these parts are shared. TSOP853.. are quite difficult to find on the internet and are not available on the most popular supply websites. TSOP62.. receivers are available at a very cheap cost but must be purchased in bulk. Single units of the TSOP75…W IR receiver are available to be purchased on popular supply websites. Along with the size and wide sensing angle and availability, the TSOP75…W is the most attractive choice of IR receiver for this project.

The remote control system for the Super-Doubler will utilize infrared communication rather than Bluetooth for its cost ease of implementation. Infrared red transmitters and receivers are very inexpensive and easy to obtain. The use of Bluetooth technology would require the acquisition of the license (which requires paying an administrative fee in order to use the brand) or would require more complex hardware to connect to devices already using the technology. The method of receiving and reading the transmitted data sent from an infrared transmitter involves less complex hardware and software with much of the software available through open sources.

3.4.15 Power

Both Linear voltage regulators and switching voltage regulators are to be considered for different sections of this project. These parts have both key features and draw backs that make them preferred for some tasks and undesirable for others. When supplying DC power to electronics and integrated circuits, parameters such as dropout voltage, transient response, power consumption and heat dissipation must be accounted for when choosing the type of voltage regulator needed to supply power to the part.

3.4.15.1 Power Requirements

The Super-Doubler will utilize both analog and digital components. Many of these components require several power sources to operate. The FPGA, ADC and MCU will all need several voltage sources to power their analog and digital systems. Since digital systems draw current in surges, effectively creating noise, the analog supplies will be taken from a separate supply rail to limit the noise as seen by the more sensitive analog systems.

Systems requiring low noise, such as the digital cores of the FPGA and MCU, will receive power regulated by low drop-out regulators. Since the supply voltages and currents needed to power these processors are relatively low and thanks the efficiency of modern LDO's, these linear regulators can be used without the need for heat sinks. Many of these surface mount parts contain a thermal plate on the bottom side of the package to help dissipate heat created by the regulator. Analog supplies will be delivered by high efficiency synchronous buck converters. Since modern switching converters produce an output voltage with relatively low noise, they can confidently be used, along with noise filtering capacitor networks and other function enabling components, to supply power to the analog supplies of the system's processors. It is important to use switching regulators when possible to maintain overall system power efficiency as well as to maintain a low system operating temperature. These regulators also contain a thermal pad on the under-side of the package the help dissipate heat away from the electronics.

Switching regulators will also be used to power other active devices that are insensitive to small voltage ripple from the power supply. The LMZ10500 simple switcher is an excellent choice for providing power to these components because of its size and ease of implementation.

Synchronization of the switching voltage regulators is also desired to help prevent unwanted noise in the power distribution system. There are several switching regulators that are able to be synchronized to another regulator or an external clock. Synchronization of the switching voltage regulators eliminates unwanted noise and the occurrence of beat harmonics produced by the internal switching actions of the regulators.

There are several design considerations that must be accounted for when designing the power distribution system. To begin, the main power supply used in this project will be taken from a wall outlet and rectified using an AC to DC adapter. This rectified voltage is not quite ready to use as a supply to components in the system. The adapters output will contain some ripple voltage that must be controlled. The single output voltage will also need to be stepped-down to provide specific supply voltages to individual or groups of parts. The power supply must be free of ripple voltage or as close to free as possible; some parts are insensitive to slight ripple voltage while other parts require a supply have a ripple voltage no greater than five percent of its nominal voltage value.

Voltage regulators are perhaps the most important part of the power distribution system. Regulators further step down the main supply voltage and compensate for much of the noise produced. Different types of regulators work differently and have different effects on power system. Regulators must compensate for changes in load resistances and currents created by internal switching and other operations of the devices being powered. When the load resistance changes, the output voltage will either sag (decrease) or swell (increase) in response forcing the regulator to compensate by increasing or decreasing the output current, respectively. This compensation, however, does not happen instantaneously so there is a significant amount of time that elapses between the initial change in load resistance the moment when the output voltage is corrected. This creates noise that must

be reduced in order to maintain system stability. Decoupling capacitors are used in parallel with the supply to limit this ripple voltage and produce a consistent power supply. The capacitor value is dictated by the frequency of the ripple voltage; Larger capacitors will be used to reduce lower frequency noise (typically generated when large numbers of components in the circuit are turned on or enabled) while smaller capacitors will be used to reduce high frequency noise (typically produced by the internal switching of digital devices). These decoupling capacitors are effective in kilohertz to hundreds of megahertz range but still do not instantly accommodate for the change in demand for current.

Although it is important and necessary to use decoupling capacitors to reduce noise in the power system, they contain parasitic properties that hinder the immediate change in supplied output current leading to the creation of noise. Inductance is the production of electromotive force in a conductor from a change in current. When the demand for current suddenly increases, the decoupling capacitors at the output of the regulator discharge to compensate for the demand. But, the capacitors do not discharge immediately do to their own internal parasitic inductances allowing for small ripple voltage. Inductances exist in capacitors, PCB traces and planes, and virtually every point of contact between a circuit component and the PCB and must be managed to maintain system stability.

3.4.15.2 Regulators and Sequencers

LINEAR REGULATORS

The voltage regulator's transient response refers to the regulator's action of adjusting the output voltage when the load current (or load resistance) suddenly changes. When these changes occur, the output voltage either sags (decreases) or swells (increases) and must be "regulated" or brought back to its nominal output value. These effects are collectively known as voltage ripple. When supplying power to FPGA's and other processor's digital cores, the ripple voltage amplitude may not exceed a certain percentage of the nominal regulated supply voltage in order to maintain system reliability and proper functionality. Modern linear voltage regulators control ripple voltage very well, many of which only allowing for a ripple amplitude in the millivolt to microvolt range. Linear regulators are a very attractive option when supplying voltage to systems that require minimum supply voltage noise.

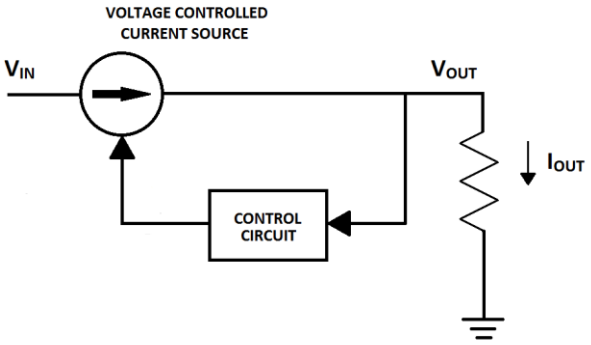


Figure 28: Diagram for linear regulator.

Figure 23 models the operation of linear voltage regulators. When the sense/control circuitry detects a drop in output current, it provides a boost in current to compensate. Often times, the sense/control circuitry consists of a difference amplifier and transistors connected in a current source configuration. The difference amplifier is connected to a reference voltage and the output voltage. As the output voltage fluctuates, the difference amplifier outputs the appropriate signal to the transistor current source to accommodate the changes.

Dropout voltage of a voltage regulator refers to the minimum voltage drop across the regulator while the part maintains a constant regulated output voltage. In other words, the drop out voltage is the minimum difference between the input voltage and output voltage required for the regulator to maintain the desired output voltage. Low dropout regulators (LDO's) offer a very low dropout voltage which will reduce the regulators consumption of power while standard linear regulators tend to have higher dropout voltages and consume more power.

Power consumption and heat dissipation must also be considered when designing the power system. Linear voltage regulators provide very consistent DC output voltages but they also dissipate a lot of heat due to the fact they are constantly "on". Heat is the effect of energy loss which is undesirable especially if the system draws power from a Battery. If linear voltage regulators are used, heat sinks may have to be implemented to help dissipate heat away from the electronics. Heat transfer must be considered to maintain a reliable power system and ensure the safety of the electronics.

TEXAS INSTRUMENTS TPS7A8101

The TPS7A8101 is a low noise LDO that provides an output voltage in the range of .8V to 6V and a maximum output current of 1A. This regulator boasts excellent noise immunity and transient response to changing output conditions. The small 3mm by 3mm package and use of few external components for stability and output voltage setting occupy very little space on the PCB reducing cost. The output voltage is set by simply adjusting the values of two resistor making this regulator very simple to use. The part costs $2.88 and is readily available at TI.com and other popular supply websites.

TEXAS INSTRUMENTS LP3878-ADJ

The LP3878-ADJ is a low-noise adjustable LDO voltage regulator for low voltage applications; 1 volt to 5 volt range. This voltage regulator, manufactured by Texas Instruments, is designed for applications involving the use of ASIC power supplies in computers and graphics cards as well digital signal processors and FPGAs. The LP3878's input supply voltage can be anywhere from 2.5 volts to 16 volts and can output voltages from 1 volt to 5.5 volts as adjusted by the designer.

The ground pin current is a low 5.5 milliamps while the output current is at the maximum 800 milliamps making this regulator very power efficient. The LP3878 produces very low

output noise typically of about 18 microvolts. Other than its notable DC characteristics, this surface-mount package is very simple and requires very little external components to help manage the voltage and noise regulation. This part costs $0.96 per unit when bought individually

ANALOG DEVICES ADM7171

The ADM7171 is a low drop-out linear voltage regulator manufactured by Analog Devices that provides an output voltage of $V_{IN} – V_{DO}$. The input voltage ranges from 2.3 volts to 6.5 volts and the drop-out voltage (V_{DO}) is advertised to be 42 millivolts when the output voltage is set to 3 volts. This regulator provides very low noise of about 5 microvolts limiting the need for external noise filtering and is independent of the output voltage. There are several fixed output voltage options for this regulator further limiting the need for external circuitry. The ADM7171 comes in a 3mm x 3mm surface mount package containing a thermal pad for increased thermal performance and protection when operating at the maximum 1amp output. Another notable feature of this LDO is the fast transient time of 1.5 microseconds for load currents ranging from 1 milliamp to 500 milliamps. This regulator is recommended for applications utilizing Communications, ADC and DAC, and phase locked loops and voltage controlled oscillators. The ADM7171 is an attractive option for supplying power to the ADC used in this project. However, the cost of his part is rather high at $3.09 per unit if bought individually on popular supply websites.

Advantages of using linear regulators include a very consistent DC voltage output and ease of design and implementation. Modern linear regulators offer a supply voltage containing a very small ripple voltage (often in the millivolt to microvolt range) minimizing the need for external filtering. When supplying power to the FPGA digital core, noise must be minimized to maintain system reliability. Since the auxiliary power supplies of the FPGA require such a low supply voltage, it may be advantageous to use a LDO to minimize power consumption and ripple voltage.

SWITCHING REGULATOR

Switching regulators produce a DC voltage supply much more efficiently in terms of power consumption than linear voltage regulators. Switching regulators are also smaller in size and their utilization leads to using less room on the printed circuit board. There are several types of switching regulators including Boost, Buck, Buck-Boost, and Fly-back. Boost regulators are DC to DC regulators that output a voltage higher than the input voltage. Buck regulators are another DC to DC converter that steps the input voltage down to pass at the output. Buck-Boost regulators are regulators that supply a voltage opposite in polarity to the input voltage. The Fly-back voltage regulator offers the use of multiple voltage output levels in both the same and opposite polarities of the input. Since this project only requires the stepping down of a DC input, the buck regulator will be the focus for researching switching voltage regulators.

The image above shows a simple model of the buck converter. A transistor is used as the switching device utilizing pulse width modulation. When the switch is on, the input voltage

is supplied to the inductor forcing a voltage drop across it ($V_{in} – V_{out}$) allowing the inductor current to increase. Once the switch is turned off, the current will decrease and the voltage drop across the inductor will switch polarities biasing the diode, the inductor and capacitor current flows through the load and back into the diode.

The idea is to maintain (as closes as possible) direct current through the load which yields a consistent and constant voltage output. As the load current decreases more drastically, the switch will stay on (or pulse) longer compensating for the larger loss in output current.

The advantages of switching voltage regulators are energy efficiency (many regulators operating at 80 to 90% efficiency) and physical size. However, these devices produce more noise than linear regulators due to the internal switching operation. A capacitor network is needed to filter out the unwanted ripple voltage from the output rail prior to being connected to the load. Controlling the noise is very important when providing power to devices such as FPGA's that require low noise power supplies. This requires much more design than circuits that utilize linear regulators.

TEXAS INSTRUMENTS TPS40305 SYNCHRONOUS BUCK CONTROLLER

The TPS40305 is a synchronous buck controller that provides a programmable output voltage from an input of 3V to 20V. This controller provides a high output current of 10A which would be useful to power multiple systems. A controller is preferred over a converter or regulator to produce high output current since the current is handled by the external parts. The overall cost is less for the controller circuits versus converters with this output current capability which tend to cost several dollars. The TPS40305 costs $2.75 and is readily available at TI.com and other popular supply websites.

TEXAS INSTRUMETNS TPS54227 SYCHRONOUS STEP DOWN CONVERTER

The TPS54227 is a synchronous buck converter manufactured by Texas Instruments that utilizes a wide input voltage range of 4.5 volts to 18 volts. This switching regulator delivers a programmable output voltage in the range of .76 volts to 7 volts and is optimized for lower duty cycle applications. Typical applications for this regulator include digital television power supplies and Blu-ray disc players. With a switching frequency of 700kHz, The TPS54227 produces low output ripple that can be filtered using a ceramic capacitor. It also includes an adjustable or pre-biased soft start. The enable pin allows this device to be connected to and controlled by a sequencer for use with systems requiring sequenced power turn-on and off. Overall surface area of this device is 5.8mm by 5mm which includes the lead length.

TEXAS INSTRUMENTS TPS56x200 SYNCHRONOUS STEP-DOWN

The TPS56x200 family of synchronous buck converters offer a 2 amp or 3 amp converter in a low profile 1.6mm X 2.9mm SOT package. These converters utilize a wide input voltage range of 4.5 volts to 17 volts and output a programmable voltage supply of .76 volts to 7 volts. Some notable features of these devices include fast transient response, low

shutdown current of less than 10 micro amps and Advanced Eco-mode pulse-skip. This advanced eco-mode operation is designed to maintain high power efficiency in light load operation. These devices are designed to minimize the use of external components for filtering ripple voltages and achieve a low standby current. This regulator costs $1.56 when purchased individually on popular supply websites

TEXAS INSTRMENTS LMZ10500 SIMPLE SWTCHER

The LMZ10500 is a switching voltage regulator that operates from a 2.7 volt to 5.5 volt input and allows for a maximum output current of 650 milliamps and an output voltage of .6 volts to 3.6 volts with low noise and minimal use of external filtering components. This device is advertised to operate at 95% power efficiency greatly reducing the generation of heat. This particular regulator is ideal in situations where space is very limited with surface area of 3mm x 2.60mm and the need for few filtering components. Though very small and simple, this device provides several safety features such as thermal shutdown protection, current limiting, and integrated compensation for stability. These Simple Switchers are available on popular supply websites and cost $2.43 per unit when purchased individually

COMPARISON

The switching regulators mentioned in the previous sections are all valid considerations for the power distribution system in the Super-Doubler. The deciding factors in choosing the appropriate parts are cost and availability, ease of implementation and complexity, and consideration of the systems being powered. Some systems will require an extremely low amount of noise that switching regulators cannot provide while others will be insensitive to ripple voltages electing the more efficient switching regulators as the prime candidate. The TPS54227, LM2653 and the LMZ10500 are the least complex of the switching regulators discussed; the least complex being the LMZ10500 simple switcher. The TPS7A8101 is ideal for powering FPGA’s, ADC’s and MCU’s since they can be programmed to provide a wide range of output voltages and can be sequenced to allow for sequenced power on and off: FPGA’s, ADC’s and MCU’s utilizing multiple power rails.

Many factors such as cost, efficiency and size play important roles in the process of choosing the parts needed for the power distribution system. While linear regulators provide the most consistent DC voltage supply, they are the most inefficient in terms of power consumption and heat dissipation. Switching regulators are the most energy efficient regulators but do not provide the most consistent DC voltage supply; containing a significant ripple voltage that requires external parts to control. The use of external parts increases the space needed to implement the circuit on the PCB as well as electro-magnetic interference existing within the system. The features and drawbacks of each type of regulator must be weighed when designing the power system for specific electronic parts. FPGA’s, ADC’s and MCU’s require low digital core voltage supplies that contain minimum noise. Since these voltages are required to be as clean as possible, LDO’s may be the desired regulators to supply these voltages since they provide a consistent voltage level.

SEQUENCER

Because they operate off of more than one voltage supply rail, FPGA's, ADC's and MCU's require voltage supply sequencing when powering up and down to ensure the safety of the processors. Data sheets of the parts used in this project provide the proper order for voltage supplies to power on and off (typically, the power-off order is exactly opposite of the power-on order). To achieve a structured power-on order for voltage supplies, a sequencer is used in the power circuit. The sequencer outputs flag signals that connect to regulator IC enable inputs that basically tell the regulator when to turn on. Sequencers may also control the voltage ramp-up time for each regulator. Texas Instruments offers several solutions for sequencing power supplies in the form of integrated circuits.

TEXAS INSTRUMENTS LM3880 POWER SEQUENCER

The LM3880 is a 3 output device that sequences the power on and off of power supplies delivered by voltage regulators. This device connects to up to 3 independent regulators at their enable inputs and turns them on in a staggered fashion. This device runs of 2.7 volts to 5.5 volts and has a low quiescent current of 25 micro-amps.

The order of which the flags power on the regulators is flag 1, 2 and 3 and is reversed when powering down as shown in the graphic above. The time between the rising edge of each flag output is equal to 16 milliseconds. This sequencer is very compact and preserves space on the printed circuit board: an area of 2.90mm x 1.6mm.

Table 13 shows the pin layout of the LM3880 sequencer. The sequencer is a very simple 6 pin device that contains an input voltage pin, ground pin, enable pin and three output flag pins that connect to the voltage regulators to be sequenced. This sequencer costs $1.38 when purchased individually on popular supply websites.

Table 13: Pinout of LM3880 power sequencer.

PIN		DESCRITION
NUMBER	NAME	
1	VCC	Input voltage
2	GND	Ground
3	EN	Enable
4	FLAG 3	Output 3 to Regulator Enable
5	FLAG 2	Output 2 to Regulator Enable
6	FLAG 1	Output 1 to Regulator Enable

4. Design

The following section details the proposed design for the hardware and software configuration of the system.

4.1 Hardware

4.1.1 Video Decoding Subsystem

The video decoding subsystem includes the ADV7181 and adds additional filtering and power circuitry. Video subsystems are high frequency systems that are sensitive to noise and interference. To address these issues we incorporate several features important to these types of systems.

To address issues of power supply noise, we employ decoupling capacitors throughout. However, there are some specific issues relating to the ADV7181. As recommended by the manufacturer, in order to avoid resistive vias and maximize their effectiveness, decoupling capacitors should be placed on the same side of the board as the ADV7181, within 0.5 cm from each power pin on the PCB, and flow should go from the power plane to the capacitor and then end at the power pin. As shown in Figure 29 below, we can see that it indicates the recommended power supply coupling.

Limiting noise migration from the digital circuitry into the analog circuitry is also important in sensitive video systems. To contain digital noise, we create a ground plane with a space between the analog and digital circuitry of the PCB board.

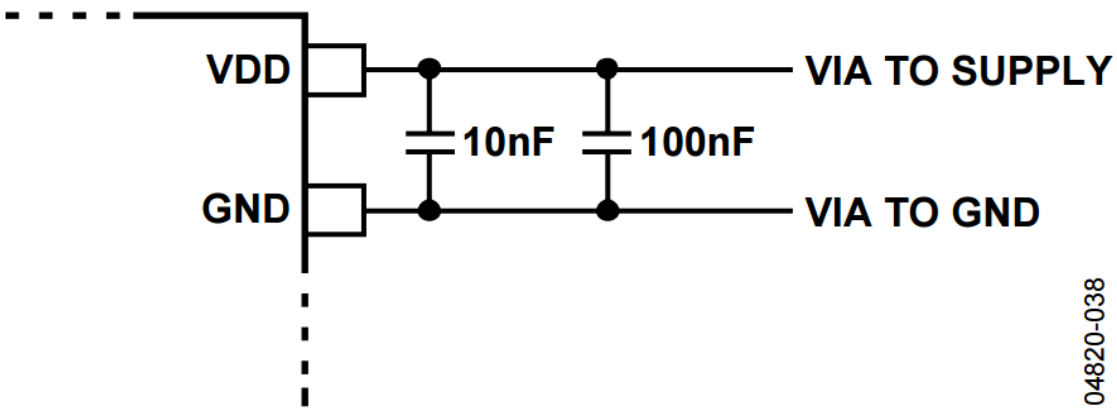


Figure 29: Power coupling. Courtesy of Analog.com.

Finally, aliasing in a digital system occurs when the input signal exceeds the Nyquist frequency. In order to avoid aliasing in our system we employ a lower pass filter on each analog input reducing aliasing noise. The antialiasing/low pass filter we chose to use is the THS7353. Recommended circuit to connect the ADV7181 and low pass filter is shown below. The THS7353 provides I2C control of all of its features. Specifically, different filters can be selected which may be necessary for some types of inputs.

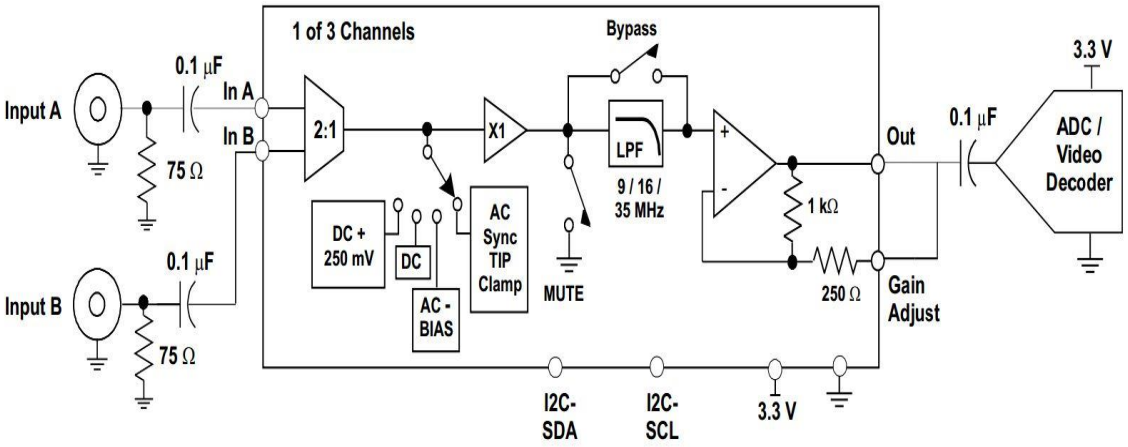


Figure 30: Low pass filter circuit with ADV7181. Texas Instruments Permission pending.

4.1.2 Audio Decoding Subsystem

As with the video decoding system, the audio system must be built to be resilient against noise. With the AD1871 It is important to configure analog and digital components of the decoder separate to limit coupling of the signals. As with the video system, while the ground plane is common to both the digital and analog sides. It is constructed of two largely separated sections. These two sections should be connected at a single point with a short zero-ohm resistor or ferrite bead. If necessary, the AD1871 has decimation and highpass filters to address, respectively, sample rate reduction and dc-blocking. The AD1871 functional diagram is shown below in Figure 31. Within our prototype, we plan to couple the audio decoder to the MCU using the SPI connection.

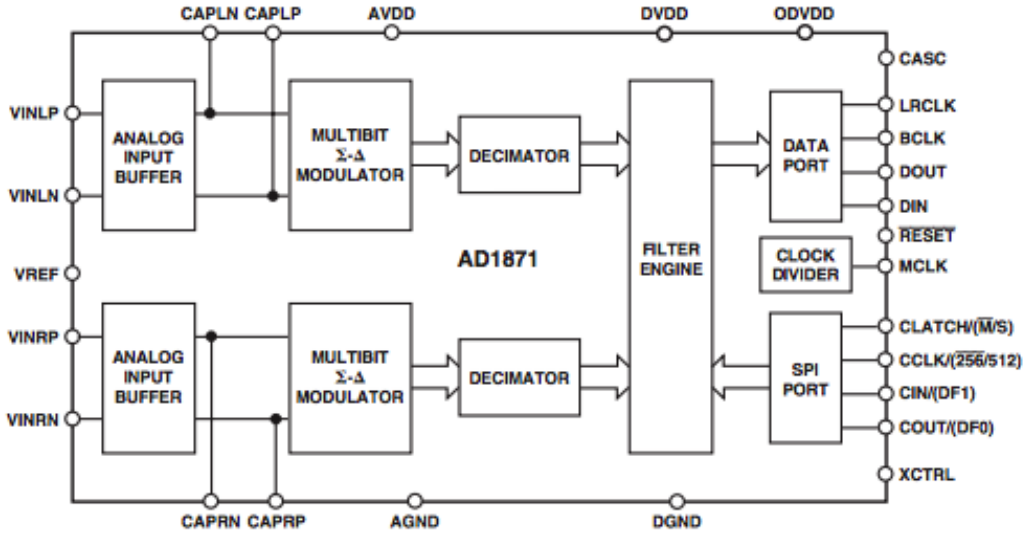


Figure 31: AD1871 functional diagram. Courtesy of Analog.com.

4.1.3 HDMI Encoding Subsystem

The HDMI encoding subsystem is responsible for converting the parallel RGB video stream being sent from the FPGA into the serial TMDS format used by the HDMI standard. This is accomplished through a unique 8-bit to 10-bit packet encoding scheme. It is also responsible for embedding the digital audio in the TMDS output. We are using Analog Device's ADV7511 HDMI transmitter to encode our HDMI video stream. The input audio format is I2S and the color space will remain RGB. The ADV7511 supports up to 36-bits of pixel resolution however we are only using 24-bits of pixel resolution. The ADV7511 will be used as a slave device on the I2C communication bus, with the microcontroller unit acting as the master. The PCB footprint is freely available from the Analog Device's website. A functional block diagram of the ADV7511 is given in Figure 32 below.

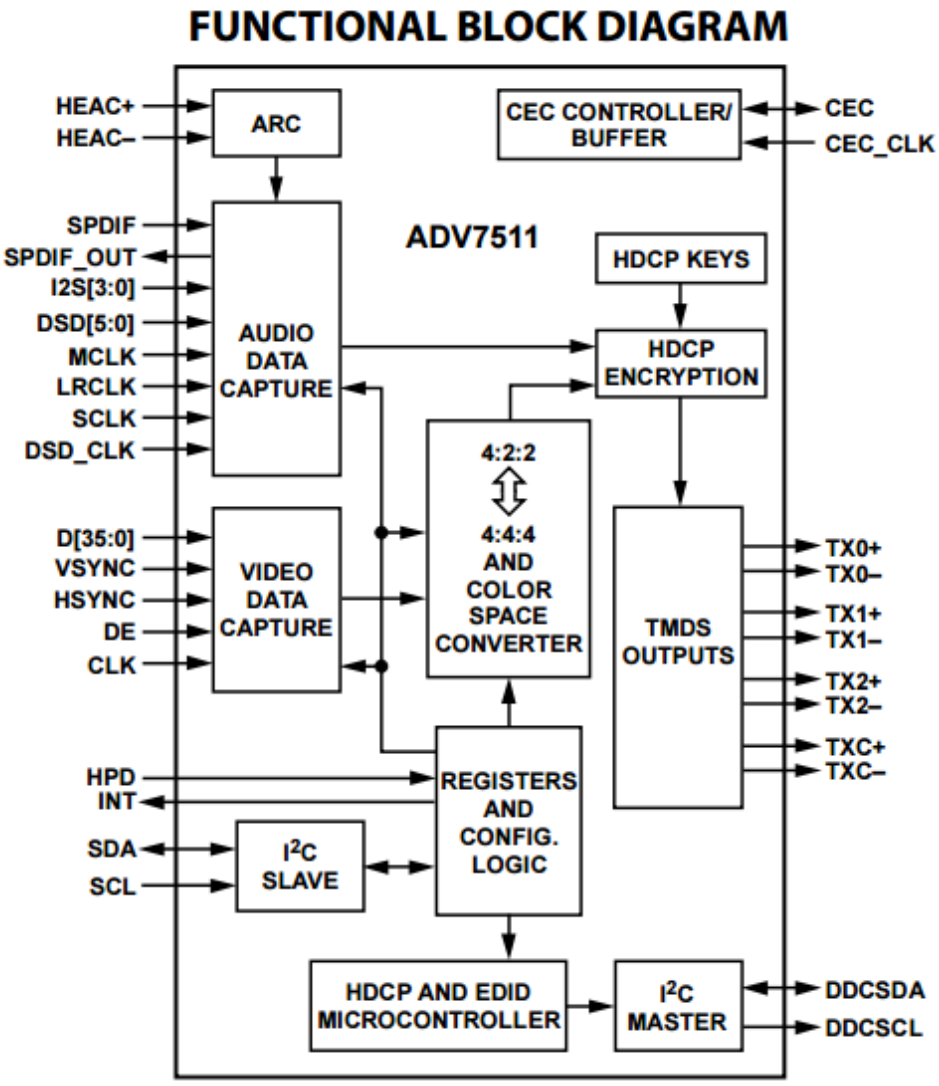


Figure 32: ADV7511 HDMI transmitter Functional block diagram. Analog Devices permission pending.

The Kintex-7 KC705 development board from Xilinx features an HDMI source which is driven by an ADV7511 integrated circuit. The University of Central Florida provided our group with a KC705 so that we could learn how to use the ADV7511 with a proven physical design. We will test the ADV7511 with our Verilog code by using the KC705 development board. The design software being used is Xilinx's Vivado Design Suite.Later in the development cycle a custom ADV7511 daughter card for the ARTY development board will be used to verify our physical design.

The test pattern shown in Figure 33 below will be used to verify that the HDMI output is working correctly. It will also be used to adjust signal levels to provide acceptable color levels. The first stage of testing will be to use HDMI to synchronize with the display with no color information. Once we can synchronize to the display we will begin testing video data transmission; we will first test standard definition output and once that is verified we will move to a higher definition output.

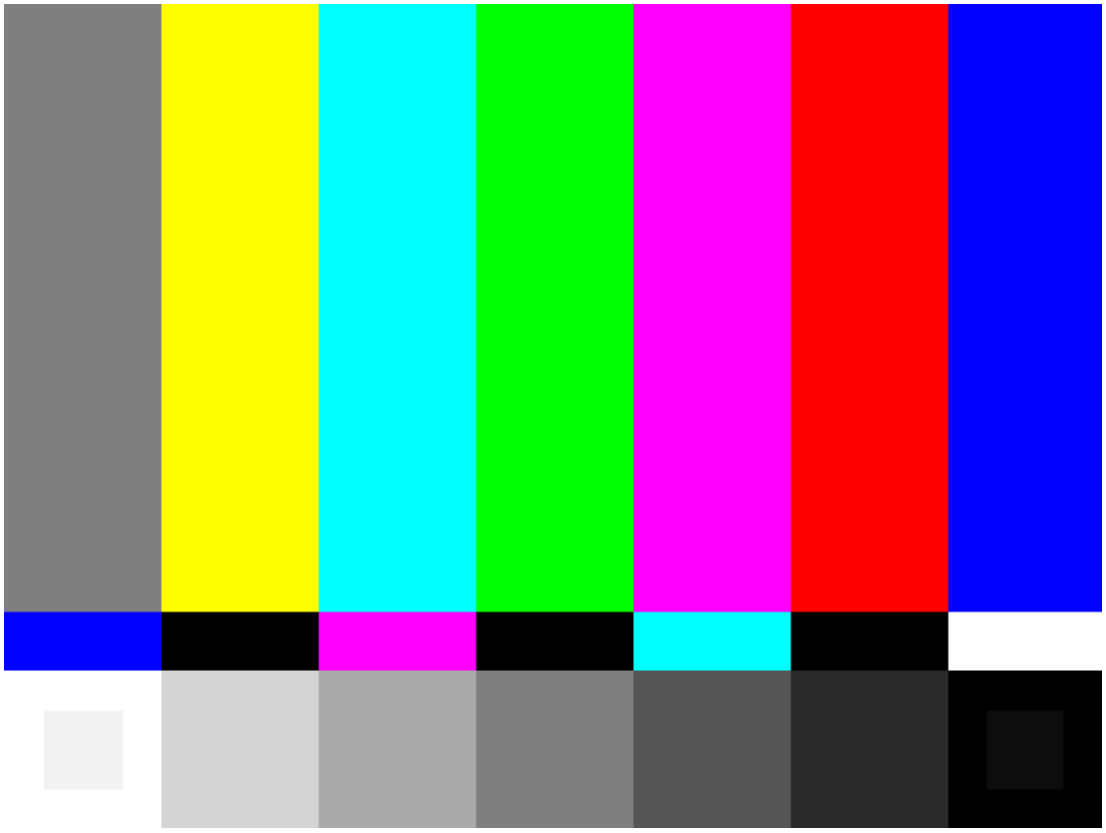


Figure 33: This test pattern will be used to verify that the digital video output is functioning correctly.

4.1.4 VGA Output Stage

As previously mentioned in Section 3.4.12 VGA Output, our video DAC of choice will be a discrete R2R ladder. The low cost and simplicity of this DAC make it an excellent choice for our system. A simple pass-through of the input audio will be used to provide accompanying analog audio to the VGA output. As shown in Figure 34, a set of buffers will be used on the analog audio and video outputs to help maintain signal strength in the presence of a load.

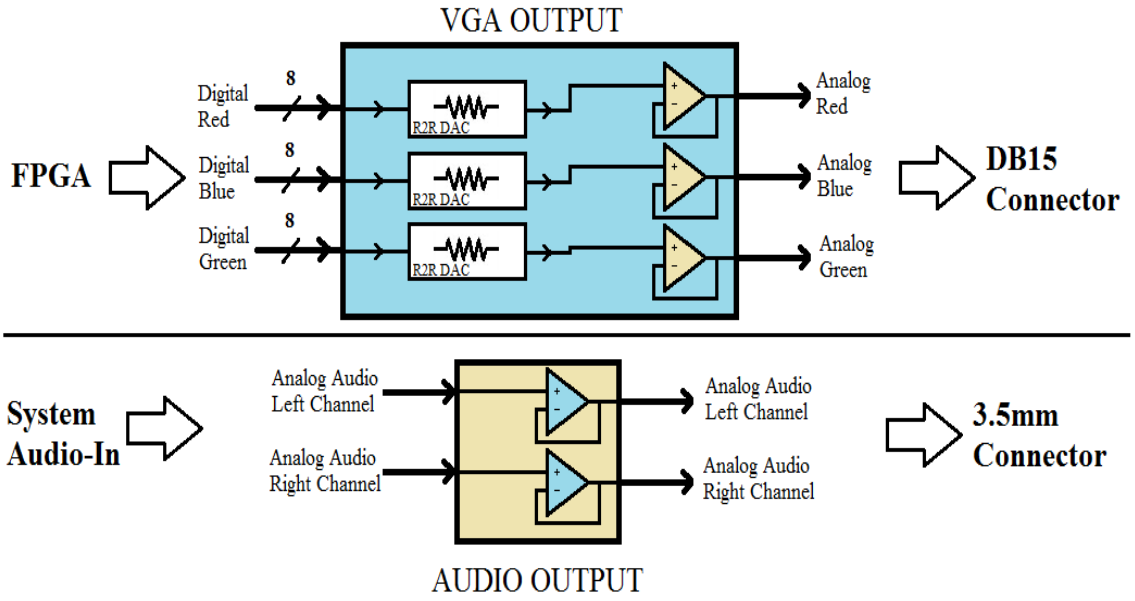


Figure 34: Functional diagrams for analog audio-video output subsystem. Synchronization signals not shown.

A total of 26 GPIO pins will be used on the FPGA for the VGA output, 24-bit per pixel, horizontal synchronization, and vertical synchronization. The three 8-bit data signals will be directly input to three R2R DACs. The output of the R2R DACs will be fed into a set of buffers to prevent the monitor's input impedance from loading the R2R DAC. The LMH6645MF operational amplifier from TI was chosen for its rail-to-rail operation, unity gain stability, and low-power consumption. A separate LMH6645MF integrated circuit will be used for each DAC to minimize crosstalk. Resistors with a tolerance of 1% will be used to minimize error in the R2R DAC. The three analog RGB signals along with the two synchronization signals form a complete VGA output; a female DB15 connector will be used as the analog video output port.

Analog audio will be provided alongside the VGA video port to form a complete analog AV output. The stereo audio signal will be taken from the Super Doubler's AV input ports thus remaining in the original audio format. Two LMH6645MF operational amplifiers will

be used to buffer the audio signals prior to being routed to a 3.5mm stereo audio connector. A brief breakdown of component quantities is given in Table 14 below.

Table 14: Components required for analog AV output subsystem.

Analog AV Subsystem Component Breakdown			
Component	Quantity	Package	Price ($)
Resistor	48	1206 (3216 metric)	13.92
OPAMP (LMH6645MF)	5	SOT-23	2.25
DB15 connector	1	N/A	2.15
3.5 mm audio jack	1	N/A	1.95

The ARTY development board will be used to test our VGA output. Three Pmod R2R DACs from DigilentInc, one is shown in Figure 35, will be used for the red, green, and blue video signals. The design of the VGA controller is covered in Section 4.2.1.2.2 Audio/Video Pipeline Interface. Each Pmod R2R DAC fits into one of the four Pmod headers on the ARTY development board. The output of each Pmod R2R DAC will be input to a LMH6645MF operational amplifier. Using trimmer potentiometers we will tune the VGA output signals. A Sony KDL-40BX420 HDTV and an ASUS computer monitor will be used to qualitatively determine acceptable signal levels based on the Super Doubler's target source content. The Sony TV and the ASUS computer monitor both feature VGA and HDMI inputs which will allow us to validate the design in several display scenarios.

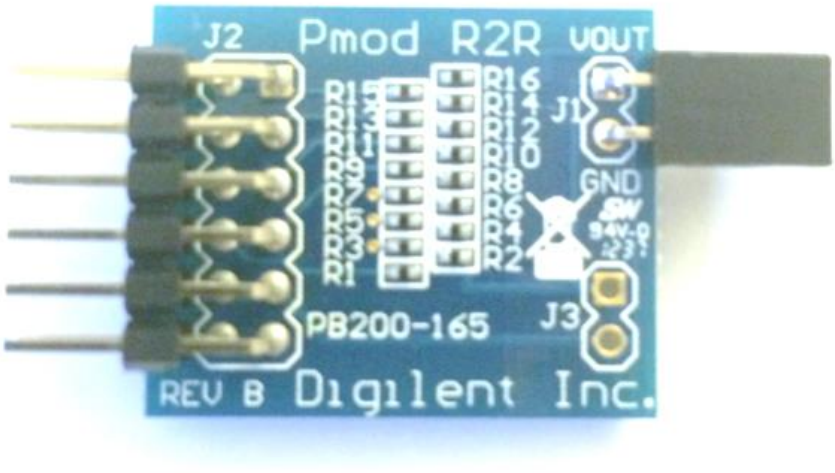


Figure 35: 8-bit R2R DAC. Will be used to test VGA output.

4.1.5 Remote Control Subsystem

The remote control that will be used for this system is a very simple remote containing 9 32-bit inputs. It sends this information over a carrier of 38kHz using an IRLED to transmit the signal. The remote control along with each of the input codes is shown in the Figure 36.

Figure 36: IR Remote Control with Input Codes. Courtesy of sparkfun.com.

The infrared signal receiver used for this project is a simple, three-terminal device that will receive infrared signal sent from the remote control. The receiver sends the received data to the MCU via one output pin for decoding. The microcontroller will read the signal and perform the programmed tasks accordingly. The TSOP75338W infrared receiver contains the photodetector, demodulator and preamplifier in the same package. This receiver demodulates signals with carriers equal to 38kHz making it compatible with the selected remote control.

The IR control kit from sparkfun.com will be purchased for testing the system. The IR receiver and photodetectors can be easily installed on a breadboard and connected to the MCU. In order to test the system, each input on the remote control must be selected and the MCU must produce an output specific to that input. LED's can be used to indicate the system is working properly.

4.1.6 LED Indicators Subsystem

The LED subsystem used is kept intentionally simple. Using only two LEDs, the requirements needed for informing the user and developer about system status can be accomplished. These LEDs will be used to indicate during normal operation MCU and FPGA communication and the status detected by the MCU of the HDMI transmitter HPD signal. During testing specific LED sequences are defined for various error or status states for quick developer evaluation. The LEDs used will be driven directly by the MCU's GPIOs and will be connected directly to ground with a series current-limiting resistor. The resistor value will be chosen according to the LED used to stay within acceptable current limits for the MCU GPIOs.

4.1.7 Subsystems Interfacing

Testing Digital Video Capture with ARTY

Due to the pin limited nature of lower-cost FPGA development boards such as the ARTY, creative methods to reduce pin usage must be implemented. The ADV7181C video decoder outputs video data with 24-bits per pixel. The four Pmod headers on the ARTY development board provide 32 GPIO pins, however the VGA test circuit uses 24 GPIO which leaves very little for the input. We chose to a use three SN74HC165 shift registers at the pixel input stage to reduce the required 24 GPIO pins down to three. Data will be parallel loaded into the SN74HC165 shift register upon every cycle of the 25 MHz pixel clock. The FPGA will use a higher frequency clock to shift the 8-bits of data before the next cycle of the pixel clock. This scheme introduces no timing delays and drastically reduces pin usage. A finite-state machine will be implemented in Verilog on the Artix-7 FPGA to convert the 3-bits at a rate of 200 MHz into a 24-bit wide data value at a rate of 25 MHz.

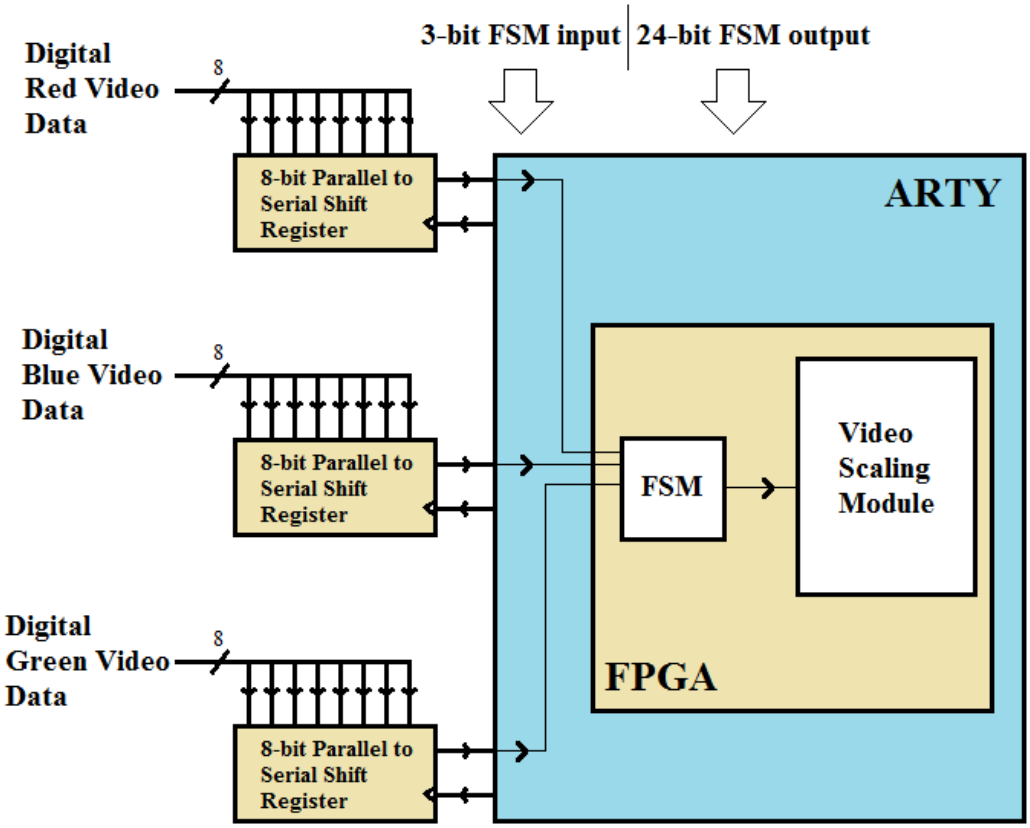


Figure 37: The external hardware used to interface the analog video decoder with the FPGA.

4.1.7.1 Microcontroller Interface

Communication between the FPGA and the microcontroller is accomplished using a 16-bit code word. Due to the GPIO pin limitations of the ARTY development board, a parallel to serial shift register will be used. Two SN74HC165 shift registers from TI will be used to realize a full 16-bit shift register. This will provide a communication path between the FPGA and microcontroller and allows the user to alter the operating mode of the system using a TV remote control.

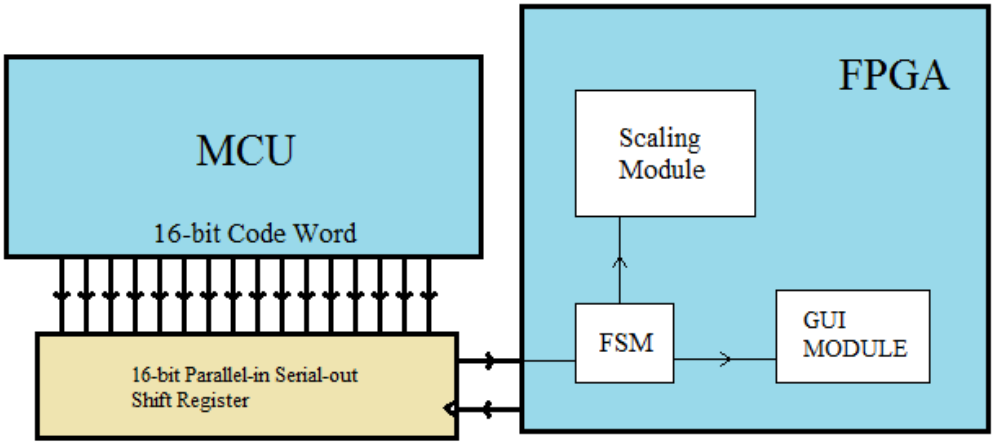


Figure 38: A shift register will provide a physical interface between the microcontroller and the FPGA.

The communication interface between the FPGA and microcontroller is used to provide the FPGA with meaningful information from the external remote control button presses. The user will have the ability to adjust various image and device settings via a GUI; settings such as brightness, color saturation, scaneline enable, and scaling factor. A code word is retrieved from the shift register and based on the OPCODE, the corresponding data within the code word will be passed to the scaling module and the GUI module. Our design does not use all of the available code words, this provides an easy way to add system functionality in future firmware updates. A detailed explanation of the code word structure and the FPGA logic interface is given in Section 4.2.1.2.1 Microcontroller Interface.

During the testing phase, an STM Nucleo will be used to place different code words in the shift register and send control flags to the FPGA. The STM Nucleo was chosen for rapid development of testing procedures due to the massive amounts of existing open-source code bases. The STM Nucleo will iterate through the various code words and we will use the on-board LEDs of the ARTY development board to confirm that the finite state machine has successfully processed the code word. Testing of the scaling module's ability to react properly to the code words is covered in Section 4.2.1.2.1 Microcontroller Interface.

4.1.8 Power Distribution System

4.1.8.1 Power Rails

The Power distribution system for the Super-Doubler will provide several different supply voltages for the required active components. Each supply will be regulated to produce the output voltages and currents to accommodate multiple loads. Though many subsystems require the same nominal supply voltage, they may not be able to share the supply rail with certain other subsystems. For this power system, supplies needed to power analog systems will be separated from supplies powering digital systems to ensure the noise generated from switching digital circuits does not affect the sensitive analog circuits. Analog and digital grounds will also be separated to further isolate analog and digital systems. Each voltage supply must contain a clean DC voltage managed by a linear or switching regulator and decoupling capacitor networks. Each device's DC characteristics and sensitivity to noise will dictate the type of voltage regulator used for the supply.

1.8V Digital Supply

The TPS563200 Synchronous Step-Down Voltage Regulator will be used to provide the 1.8V supply designated for power 1.8V digital systems. This device will be configured to provide a 1.8V output with a maximum 3A output current. Figure 39 is the reference design given on the datasheet. Some of the components in the reference design will be changed to meet the needs of this system.

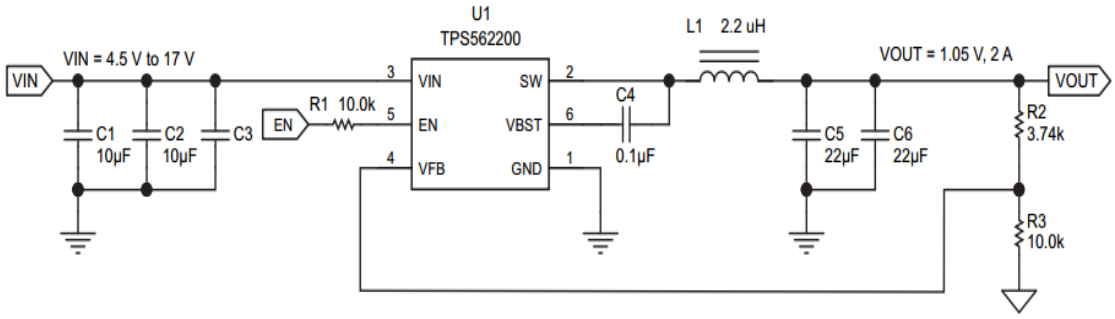


Figure 39: Reference Design for the TPS56x00. Texas Instruments permission pending.

The datasheet supplies component values needed to produce certain output voltages making the design process for this regulator very simple. Current analysis will be conducted to determine the current ratings of the circuit components. Table 15 lists the components and values determined in the design guide.

Table 15: Component values for the 1.8V regulator circuit

Output Voltage	R2	R3	L1			C5, C6, C7
			Min	Typ	Max	
1.8 V	13.7 kΩ	10 kΩ	1.5 µH	2.2 µH	4.7 µH	20 µF - 68 µF

Current ratings of the inductor and capacitors will be determined to ensure proper functionality. Beginning with the inductor; the peak-to-peak inductor ripple current, the overall peak inductor current and the RMS inductor current are calculated in Equations 1, 2 and 3.

$$I_{ptpripple} = \frac{V_{out}}{V_{in,max}} \times \frac{V_{in.max} - V_{out}}{L_1 \times f_{sw}} = \frac{1.8\,V}{17\,V} \times \frac{17\,V - 1.8\,V}{2.2\,\mu H \times 650\,kHz} = 1.125\,A \qquad (1)$$

$$I_{peak} = I_o + \frac{I_{ptpripple}}{2} = 3\,A + \frac{1.125}{2} = 3.563\,A \qquad (2)$$

$$I_{L,RMS} = \sqrt{{I_o}^2 + \frac{1}{12} I_{ptpripple}^2} = \sqrt{3A^{\,2} + \frac{1}{12} 1.125A^{\,2}} = 3.017\,A \qquad (3)$$

The components in the in regulator circuit are to be chosen with ratings that exceed those calculated in the analysis.

The RMS current for the output capacitors are shown in Equation 4**.**

$$I_{CO,RMS} = \frac{V_{out} \times (V_{in} - V_{out})}{\sqrt{12 \times V_{in} \times L_o \times f_{sw}}} = \frac{1.8 \times (12V - 1.8V)}{\sqrt{12 \times 12V \times 2.2\mu H \times 650kHz}} = 1.279A \qquad (4)$$

The parameter f_{sw} is the 650kHz switching frequency of the regulator.

The input capacitors are recommended to be greater than 10µF. Two of these capacitors are used to filter any noise at the input of the regulator. A .1µF bypass capacitor is also recommended to filter any high frequency ripple. These capacitors will need to have a voltage rating greater than the input voltage.

The Bootstrap capacitor should connect the V_{BST} pin and SW pin in order for the regulator to work properly. The datasheet recommends this capacitor be .1µF ceramic capacitor.

Upon determining the values of the components for this design, the part numbers and cost are obtained. Table 17 displays the components and other relevant information for this supply circuit.

Table 17: 1.8V Regulator Circuit BOM

Component	Value	Manufacturer	Part Number	QTY	Cost/Unit
R1	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
R2	13.7 kΩ	Vishay Dale	CRCW040213K7FKED	1	$0.10
R3	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
C1, C2	10 µF	Murata	GRM21BR61C106KE15K	2	$0.20
C3, C4	.1 µF	Murata	GRM155R71C104KA88D	2	$0.10
C5, C6	47 µF	Murata	GRM32ER61C476KE15K	2	$0.94
C7	22 µF	Murata	GRM31CR61C226ME15L	1	$0.59
L1	2.2 µH	TDK	CLF6045T-2R2N-D	1	$1.32
U1		TI	TPS563200	1	$1.58
TOTAL COST			$6.27		

1.0V Digital supply

The TPS40305 Synchronous Buck Controller will be used to regulate the 1.0V power supply. This controller provides the 1 volt ouput voltage and allows for a maximum current of 10 amps. This high current capability is necessary to accommodate the needs some FPGA supplies. The reference design for this power supply circuit is shown in Figure 40.

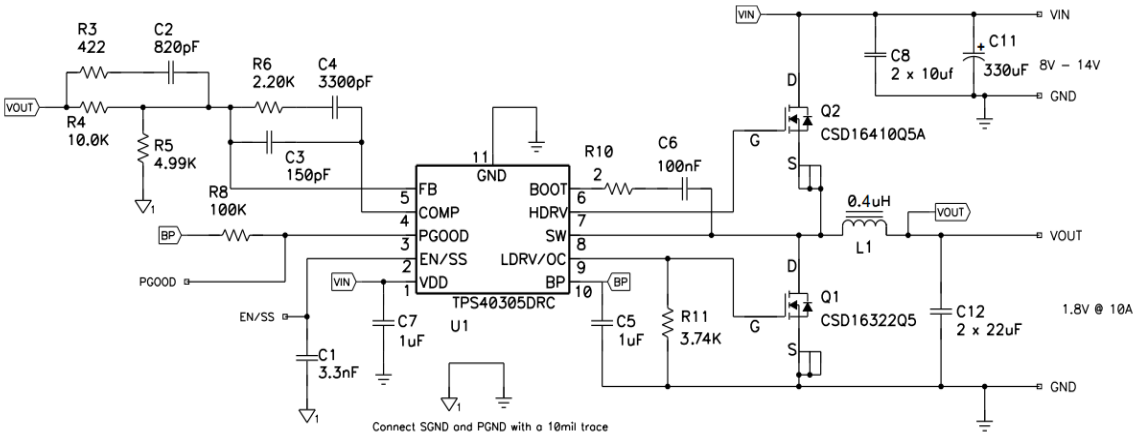


Figure 40: TPS40305 Reference design. Texas Instruments permission pending.

The datasheet for this device contains step by step procedures to design the system accordingly. To begin, the value of the inductor is determined using Equation 5.

$$L_1 = \frac{V_{in} - V_{out}}{.3 * I_{out}} \times \frac{V_{out}}{V_{in}} \times \frac{1}{f_{sw}} \tag{5}$$

V_{in} is the controllers maximum input voltage (12 volts). V_{out} is the output voltage value (1 volt), f_{sw} is the switching frequency of 1200kHz, and I_{out} is the maximum current output (10 amps). The output current is multiplied by .3 since buck power inductors are designed

to accommodate 30% ripple current. Upon entering the values listed, the equation yields approximately 255nH, A standard 240nH inductor will be used in the final design. Using this value, the output capacitance is calculated to using Equation 6.

$$C_{out} = \frac{I_{Transient}^2 \times L_1}{V_{out} \times V_{over}} = \frac{4^2 \times 240nH}{1V \times .1V} = 38.4\mu F \quad (6)$$

The maximum transient current is approximated to be 4 amps and the over-voltage is equal to .1V. Entering these values in the equation returns the output capacitance of 38.4µF. Before choosing the output capacitors, the maximum equivalent series resistance must be determined using Equation 7.

$$ESR_{max} = \frac{V_{ripple(total)} - \frac{I_{ripple}}{8 \times C_{out} \times f_{sw}}}{I_{ripple}} = \frac{20mV - \frac{3.0A}{8 \times 38.4\mu F \times 1200KHz}}{3.0A} \quad (7)$$

$$= 3.95m\Omega$$

Knowing the minimum output capacitance and the maximum ESR, the appropriate capacitors can be selected. Two 22µF low-ESR ceramic X5R-type capacitors rated for 6.3 volts will be used as the output bypass capacitors.

Next, input capacitance and maximum ESR are determined in Equations 8 and 9.

$$C_{in} = \frac{I_{load} \times V_{out}}{V_{ripple} \times V_{in} \times f_{sw}} = \frac{10A \times 1V}{.100mV \times 12V \times 1200KHz} = 4.63\mu F \quad (8)$$

$$ESR_{max} = \frac{V_{ripple(esr)}}{I_{load} \times \frac{1}{2} I_{ripple}} = \frac{150mV}{11.75A} = 12.7m\Omega \quad (9)$$

Two 3.3µF low ESR ceramic X5R-type capacitors rated for 6.3 volts will be used as the input bypass capacitors.

The MOSFETS are chosen using the Texas Instruments' NexFET MOSFET selection tool. The CSD16410Q5A and CSD16322Q5 as shown in the reference design will be used. Their current ratings of 18mA at 1200kHz make them appropriate to use in the design.

The feedback divider resistors must be determined using the internal .6 volt reference of the controller and a value for R4. As mentioned on the datasheet, R4 must be a value between 10kΩ and 50kΩ to maintain a balanced feedback current and immunity from noise. If R4 is chosen to be 10kΩ, Equation 10 is solved to obtain R5.

$$R_5 = \frac{V_{FB} \times R_4}{V_{out} - V_{FB}} = \frac{.6V \times 10k\Omega}{1V - .6V} = 15000\ \Omega \quad (10)$$

The remaining parts needed to complete this circuit are recommended by the datasheet and will be used in the final design. Table 18 displays these parts and the components previously mentioned along with relevant information for the power circuit.

Table 18: 1.0V Supply Circuit BOM.

Component	value	Manufacturer	Part number	QTY	Cost/Unit
R3	422 Ω	Vishay Dale	CRCW0402422RFKED	1	$0.10
R4	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
R5	15 kΩ	Yageo	RC0402FR-0715KL	1	$0.10
R6	2.20 kΩ	Yageo	RC0402FR-072K2L	1	$0.10
R8	100 kΩ	Yageo	RC0402FR-07100KL	1	$0.10
R10	2 Ω	Vishay Dale	CRCW0402422R00FKED	1	$0.10
R11	3.74 kΩ	Yageo	RC0402FR-073K74L	1	$0.10
C1	3.3 nf	Murata	GRM033R61A332KA01D	1	$0.10
C2	820 pF	Murata	GRM033R71E821KA01D	1	$0.10
C3	150 pF	Murata	GRM033R71E151KA01D	1	$0.10
C4	3300 pF	TDK	C0603X7R1E332K030BA	1	$0.10
C5	1.0 µF	Murata	GRM155R61A105KE15D	1	$0.10
C6	100 nF	Murata	GRM155R71C104KA88D	1	$0.10
C7	1 µF	Murata	GRM188R61E105KA12D	1	$0.10
C8	10 µf	TDK	C3216X7R1E335K160AC	2	$0.34
C11	330 µF	Nichicon	UWT1E331MNL1GS	1	$0.55
C12	22 µF	Murata	GRM31CR61E226KE15L	2	$0.56
Q1		TI	CSD16322Q5	1	$1.28
Q2		TI	CSD16410Q5A	1	$1.01
L1	0.32 µH	Vishay	IHLP-4040D7-11	1	$1.50
U1		TI	TPS40305	1	$2.75
TOTAL COST			$9.39		

1.0V Analog Supply

A 1V supply designated for analog devices is also needed in the system. However, the output current requirement is not as demanding as the digital 1V supply. Therefore, the TPS7A8101 low dropout regulator will be used to regulate the 1V analog supply. The regulator accepts a wide range of input voltage and delivers an output voltage in the range of .8V to 6V with a maximum output current of 1 amp. The provided reference material has given component values used to produce certain voltage outputs. The reference design is shown in Figure 41.

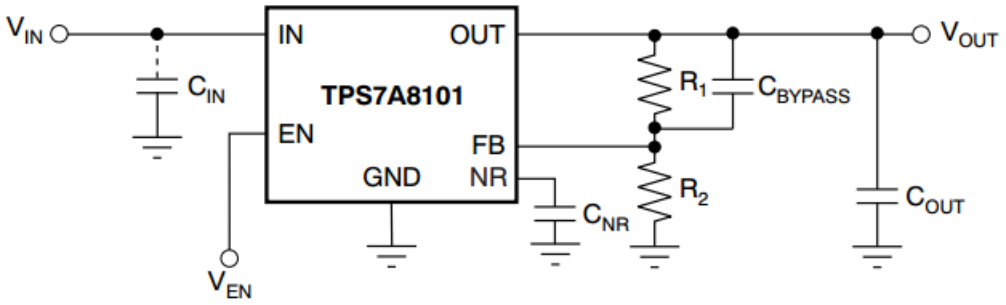


Figure 41: TPS7A8101 Reference design. Texas Instruments permission pending.

Equation 11 is used to determine the output voltage.

$$V_{out} = \frac{R_1 + R_2}{R_2} \times 0.8 \tag{11}$$

Choosing R_2 to be 10kΩ and V_{out} equal to 1V, R_1 is solved to be equal to 2.49kΩ.

Input and output capacitors are used in the design to improve noise and ripple rejection and transient response. A .1µF to 1µF ceramic capacitor is recommended by the reference guide to achieve these enhancements. The regulator is designed to be stabilized using ceramic X5R and X7R capacitors for their noise rejection capabilities and accurate ESR as temperature changes.

Table 19 displays the components and other relevant information for the final design of the 1V analog supply.

Table 19: Circuit component values for the Regulator Circuit.

Component	Value	Manufacturer	Part number	QTY	Cost/Unit
C_{in}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
C_{NR}, $C_{BYP.}$	470 nF	TDK	C1005X5R0J474K050BB	2	$0.10
C_{out}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
R_1	2.49 kΩ	Yageo	RC0402FR-072K49L	1	$0.10
R_2	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
U1		TI	TPS7A8101DRBT	1	$2.88
TOTAL COST			$3.58		

5 Volt Analog and Digital Supplies

There are both analog and digital systems that require 5 volts to operate in the Super-Doubler. Therefore, two 5 volt supplies will be designed. The TPS562300 Synchronous Step-Down Voltage Regulator will be used for both of the 5V systems. Table 20 displays component values of the voltage regulator circuit that yield a 5V output. The reference design is shown in Table 20.

Table 20: Component values for the 5V regulator circuit.

Output Voltage	R2	R3	L1			C5, C6, C7
			Min	Typ	Max	
5 V	54.9 kΩ	10 kΩ	2.2 µH	3.3 µH	4.7 µH	20 µF - 68 µF

The design methodology is identical to that of the 1.8V supply circuit since the same type of regulator is used here. The inductor current rating is determined using Equations 12, 13 and 14.

$$I_{ptpripple} = \frac{V_{out}}{V_{in,max}} \times \frac{V_{in.max} - V_{out}}{L_1 \times f_{sw}} = \frac{5V}{17V} \times \frac{17V - 5V}{3.3\mu H \times 650kHz} = 1.645A \quad (12)$$

$$I_{peak} = I_o + \frac{I_{ptpripple}}{2} = 3A + \frac{1.645A}{2} = 3.83A \quad (13)$$

$$I_{L,RMS} = \sqrt{{I_o}^2 + \frac{1}{12} I_{ptpripple}^2} = \sqrt{(3A)^2 + \frac{1}{12}(1.645A)^2} = 3.04 \quad (14)$$

The inductor current rating must be greater than 3.83A. Next, the RMS current rating for the output capacitor is determined using Equation 15**.** The capacitor must contain a current rating higher this RMS current.

$$I_{CO,RMS} = \frac{V_{out} \times (V_{in} - V_{out})}{\sqrt{12} \times V_{in} \times L_o \times f_{sw}} = \frac{5 \times (12V - 5V)}{\sqrt{12} \times 12V \times 3.3\mu H \times 650kHz} = 1.99A \quad (15)$$

Table 21 displays the components along other relevant information for the supply circuit.

Table 21: 5V Power Supply Circuit BOM.

Component	Value	Manufacturer	Part Number	QTY	Cost/Unit
R1	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
R2	54.9 kΩ	Yageo	RC0402FR-0754K9L	1	$0.10
R3	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
C1, C2	10 µF	Murata	GRM21BR61C106KE15K	2	$0.20
C3, C4	.1 µF	Murata	GRM155R71C104KA88D	2	$0.10
C5, C6	47 µF	Murata	GRM32ER61C476KE15K	2	$0.94
C7	22 µF	Murata	GRM31CR61C226ME15L	1	$0.59
L1	2.2 µH	TDK	CLF6045T-2R2N-D	1	$1.32
U1		TI	TPS563200	1	$1.58
TOTAL COST			$6.27		

Two of these circuits will be implemented on the PCB to supply 5 volts to both analog and digital systems.

1.2V Analog Supply

The TPS7A8101 Linear Voltage Regulator will also be used to regulate the 1.2V supply. The design methodology of this circuit is the same as the previous circuit employing the TPS7A8101 voltage regulator. The reference design is shown in Figure 41**.** Table 22 lists the component values and other relevant information for this voltage regulator circuit providing the 1.2V output.

Table 22: 1.2V Power Supply Circuit BOM.

Component	Value	Manufacturer	Part number	QTY	Cost/Unit
C_{in}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
$C_{NR,}$ $C_{BYP.}$	470 nF	TDK	C1005X5R0J474K050BB	2	$0.10
C_{out}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
R_1	4.99 kΩ	Yageo	RC0402FR-074K99L	1	$0.10
R_2	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
U1		TI	TPS7A8101DRBT	1	$2.88
TOTAL COST			$3.58		

3.3V Analog Supply

The TSP7A8101 linear regulator will be used to step down the 12 volt input voltage to 3.3 volts. The reference design is in Figure 41 above. The output voltage of the regulator is set by adjusting the values of R_1 and R_2. R_1 is determined in Equation 16 by setting R_2 equal to 10kΩ and V_{out} equal to 3.3V.

$$R_1 = \frac{V_{out} \times R_2}{.8} - R_2 = \frac{3.3V \times 10k\Omega}{.8} - 10k\Omega = 31 \quad (16)$$

A standard 31.2kΩ resistor will be used in the design to yield the 3.3V output. Ceramic input and output capacitors will be used to filter any voltage ripple and noise. The following table displays the circuit component values and other relevant information used in this design. The values in Table 23 were either determined given certain parameters or are suggested by the manufacturer of the voltage regulator. These capacitors are ceramic X5R ultra low ESR to improve the transient response of the system.

Table 23: 3.3V Power Supply BOM.

Component	Value	Manufacturer	Part number	QTY	Cost/Unit
C_{in1}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
C_{in2}	.1 µF	Murata	GRM155R71C104KA88D	1	$0.10
C_{NR}, $C_{BYP.}$	470 nF	TDK	C1005X5R0J474K050BB	2	$0.10
C_{out}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
C1	1 µF	Murata	GRM188R61E105KA12D	1	$0.10
C2	10 nF	Murata	GRM033R70J103kA01D	1	$0.10
R_1	30.9 kΩ	Yageo	RC0402FR-0730K9L	1	$0.10
R_2	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
U1		TI	TPS7A8101DRBT	1	$2.88
TOTAL COST			$3.98		

3.3V Digital Supply

Since 3.3V is a popular supply voltage for digital processors, the 3.3V power rail reserved for digital systems will be designed to be able to accommodate many 3.3V supplies. The TPS56300 Synchronous Step-Down voltage regulator will be used to regulate this rail at 3.3V and allow for a maximum current draw of 3A. The reference design is located in Figure 39 above. Table 24 shows the component values given in the TPS563200 user guide to produce an output voltage of 3.3V.

Table 24: Component Values for the 3.3V Regulator Circuit.

Output Voltage	R2	R3	L1			C5, C6, C7
			Min	Typ	Max	
3.3 V	33.2 kΩ	10 kΩ	1.5 µH	2.2 µH	4.7 µH	20 µF - 68 µF

Using the typical components values for this regulator circuit and the regulators switching frequency of 650kHz, the peak and RMS current through the inductor are calculated. The inductor must have a current rating that exceeds the calculated values from Equations 17, 18, and 19.

$$I_{ptpripple} = \frac{V_{out}}{V_{in,max}} \times \frac{V_{in.max} - V_{out}}{L_1 \times f_{sw}} = \frac{3.3V}{17V} \times \frac{17V - 3.3V}{2.2\mu H \times 650kHz} = 1.86A \quad (17)$$

$$I_{peak} = I_o + \frac{I_{ptpripple}}{2} = 3A + \frac{1.86A}{2} = 3.93A \quad (18)$$

$$I_{L,RMS} = \sqrt{{I_o}^2 + \frac{1}{12} I_{ptpripple}^2} = \sqrt{(3A)^2 + \frac{1}{12}(1.86A)^2} = 3.05A \quad (19)$$

The inductor current rating must be greater than 3.93A. Next, the RMS current rating for the output capacitor is determined in Equation 20**.** The capacitor must contain a current rating higher this RMS current.

$$I_{CO,RMS} = \frac{V_{out} \times (V_{in} - V_{out})}{\sqrt{12} \times V_{in} \times L_o \times f_{sw}} = \frac{3.3 \times (12V - 3.3V)}{\sqrt{12} \times 12V \times 2.2\mu H \times 650kHz} = 2.00A \qquad (20)$$

Table 25 displays the components used in this voltage regulator circuit along with other relevant information. Ceramic input decoupling capacitors with values of 10µF and .1µF are recommended.

Table 25: 3.3V Power Supply Circuit BOM.

Component	Value	Manufacturer	Part Number	QTY	Cost/Unit
R1	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
R2	33.2 kΩ	Vishay Dale	CRCW040233K2FKED	1	$0.10
R3	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
C1, C2	10 µF	Murata	GRM21BR61C106KE15K	2	$0.20
C3, C4	.1 µF	Murata	GRM155R71C104KA88D	2	$0.10
C5, C6	47 µF	Murata	GRM32ER61C476KE15K	2	$0.94
C7	22 µF	Murata	GRM31CR61C226ME15L	1	$0.59
L1	2.2 µH	TDK	CLF6045T-2R2N-D	1	$1.32
U1		TI	TPS563200	1	$1.58
TOTAL COST			$6.27		

2.5V Digital Supply

Design steps are similar to those previously discussed. With a V_{out} equal to 2..5V and R_2 equal to 10kΩ, R_1 is determined in Equation 21.

$$R_1 = \frac{V_{out} \times R_2}{.8} - R_2 = \frac{2.5V \times 10k\Omega}{.8} - 10k\Omega = 21.25k\Omega \qquad (21)$$

A standard 21kΩ resistor will be used in the voltage divider at the output. Ceramic input and output capacitors will be used to filter unwanted noise and improve transient characteristics. Table 26 displays the components and relevant information for this power regulator circuit. The MCU power rail requires an addition capacitor network connected to the power rail and ground. These capacitors are also labeled in Table 26**.**

Table 26: 2.5V Power Supply Circuit BOM.

Component	Value	Manufacturer	Part number	QTY	Cost/Unit
C_{in1}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
C_{in2}	.1 µF	Murata	GRM155R71C104KA88D	1	$0.10
$C_{NR,}$ $C_{BYP.}$	470 nF	TDK	C1005X5R0J474K050BB	2	$0.10
C_{out}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
C_1	4.7 µF	Murata	GRM188R60J475KE19D	1	$0.12
C_2, C_3	100 nF	Murata	GRM155R71C104KA88D	2	$0.10
R_1	21 kΩ	Yageo	RC0402FR-0721KL	1	$0.10
R_2	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
U1		TI	TPS7A8101DRBT	1	$2.88
TOTAL COST			$3.98		

2.7V Supply

The TPS7A8101 Linear Voltage Regulator will be used to regulate the 2.7V power rail. The reference design is in Figure 41 above. Resistors R1 and R2 set the output voltage. Equations 22 and 23 show the process for determining the value of R_1 when R_2 is set equal to 10kΩ and V_{out} is set to 2.7V

.

$$V_{out} = \frac{R_1 + R_2}{R_2} \times 0.8 \tag{22}$$

$$R_1 = \frac{V_{out} \times R_2}{.8} - R_2 = \frac{2.7V \times 10k\Omega}{.8} - 10k\Omega = 23.75k\Omega \tag{23}$$

A standard 23.7kΩ resistor will be used to deliver the 2.7V output. Input and output decoupling capacitors will be used for noise filtering. Table 27 displays the information for all components used in this voltage regulator circuit. The values in the table were either determined given certain parameters or are suggested by the manufacturer.

Table 27: 2.7V Power Supply Circuit BOM.

Component	Value	Manufacturer	Part number	QTY	Cost/Unit
C_{in1}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
C_{in2}	.1 µF	Murata	GRM155R71C104KA88D	1	$0.10
$C_{NR,}$ $C_{BYP.}$	470 nF	TDK	C1005X5R0J474K050BB	2	$0.10
C_{out}	10 µF	Murata	GRM21BR61C106KE15K	1	$0.20
R_1	23.7 kΩ	Yageo	RC0603FR-0723K7L	1	$0.10
R_2	10 kΩ	Yageo	RC0402FR-0710KL	1	$0.10
U1		TI	TPS7A8101DRBT	1	$2.88
TOTAL COST			$3.78		

All input capacitors in the two designs are ceramic X5R low ESR surface-mount capacitors rated for 25 volts. The output capacitors are ceramic X5R and X7R low ESR surface-mount capacitors rated for 6.3 volts. The resistors in these designs are 1/16 Watt chip resistors rated to be within 1% of the nominal resistance value.

The processors, decoders and other active parts will receive power from these voltage supplies. The supplies are designated to supply either analog or digital systems. Addition filtering may be required for some supplies. The input voltage of the TPS7A8101 regulators will be taken from the 5 volt power rail.

4.1.8.2 FPGA

There are several voltage supplies needed to power the digital and analog systems of the FPGA. Table 28 shows the minimum, maximum and typical voltage values for each supply needed for the FPGA.

Table 28: FPGA Power Supplies.

Supply	Minimum (V)	Maximum (V)	Typical (V)
VCCINT	.95	1.05	1.00
VCCAUX	1.71	1.89	1.80
VCCBRAM	.95	1.05	1.00
VCCO	1.14	3.465	1.20
VCCBATT	GROUND		
VMGTAVCC	.97	1.03	1.00
VMGTAVTT	1.17	1.23	1.20
VCCADC	1.71	1.89	1.80

When supplies share the same nominal value they should share the power rail. V_{CCINT} is the internal supply voltage for the FPGA logic and V_{CCBRAM} is the supply for the FPGA's block ram memories. Since these two supplies have the same nominal voltage value, the two rails will be supplied by the same regulator and will be powered on and off simultaneously. These supplies will be taken from the 1.0V digital supply rail.

In addition to the output bypass capacitors of the voltage regulator, the FPGA requires an addition noise filtering network on each power rail as referenced from Xilinx 7 Series FPGA Power Management Guide. The capacitors will be placed as close to possible to the supply pins as possible to prevent oscillation and reduce noise. Table 29 displays the additional capacitors required for each voltage supply along with other relevant information needed to complete the circuit.

Table 29: Required capacitors for V_{CCING} and V_{CCBRAM} supplies.

	Value	Manufacturer	Part number	QTY	Cost/unit
V_{CCINT}	100 µF	Murata	GRM31CR60J107ME39K	1	$0.81
	4.7 µF	Murata	GRM188R60J475KE19D	2	$0.12
	.47 µF	TDK	C1005X5R0J474K050BB	3	$0.10
V_{CCBRAM}	47 µF	Murata	GRM32ER61C476KE15K	1	$0.94
	.47 µF	TDK	C1005X5R0J474K050BB	1	$0.10
TOTAL COST			$2.07		

V_{CCO}, V_{CCAUX}, $V_{CCAUXIO}$ and V_{CCADC} supplies will be taken from the 1.8V Digital power rail. The FPGA requires an addition noise filtering network on the V_{CCO} and V_{CCAUX} supplies as referenced from Xilinx 7 Series FPGA Power Management Guide. Table 30 displays the additional capacitors required for each power rail along with other relevant information.

Table 30: Required capacitors for the V_{CCO} and V_{CCAUX} supplies.

Required Capacitors for V_{CCO}, V_{CCAUX}, $V_{CCAUXIO}$ and V_{CCADC} Power Rails					
	Value	Manufacturer	Part number	QTY	Cost/unit
V_{CCO}	100 µF	Murata	GRM31CR60J107ME39K	1	$0.81
	47 µF	Murata	GRM32ER61C476KE15K	1	$0.94
	4.7 µF	Murata	GRM188R60J475KE19D	2	$0.12
	.47 µF	TDK	C1005X5R0J474K050BB	4	$0.10
V_{CCAUX}	47 µF	Murata	GRM32ER61C476KE15K	1	$0.94
	4.7 µF	Murata	GRM188R60J475KE19D	1	$0.12
	.47 µF	TDK	C1005X5R0J474K050BB	2	$0.10
TOTAL COST			$3.65		

$V_{CCMGTAVCC}$ and $V_{CCMGTAVTT}$ are the supplies for the GTP transmitter and receiver and their termination circuits, respectively. $V_{CCMGTAVCC}$ will be taken from the 1V analog power rail and $V_{CCMGTAVTT}$ will be taken from the 1.2V analog power rail. These analog supplies may require additional decoupling capacitors placed near the supply pins of the FPGA.

4.1.8.3 MCU

The STM32F070RB microcontroller will need two power supplies in order to operate; the voltage supply (V_{DD}) for the internal regulator and I/O's and the voltage supply (V_{DDA}) for the ADC, PLL, Reset blocks, and RC's. V_{DDA} must be powered on before V_{DD} using a sequencer so the two supplies will be taken from different power rails. V_{DD} is required to be in the range of 2.4V and 3.6V while V_{DDA} is required to be in the same range as long as it is greater than V_{DD}. For this system, V_{DD} will receive power from the 2.5V power rail and V_{DDA} will be connected to the 3.3V analog supply.

The MCU requires additional decoupling capacitors placed near the supply pins. Table 30 shows these capacitors and other relevant information needed to complete the circuit.

Table 31: Required capacitors for V_{DDA} and V_{DD} supplies.

Supply	Value	Manufacturer	Part number	QTY	Cost/Unit
V_{DDA}	1 µF	Murata	GRM188R61E105KA12D	1	$0.10
	10 nF	Murata	GRM033R70J103kA01D	1	$0.10
V_{DD}	4.7 µF	Murata	GRM188R60J475KE19D	1	$0.12
	100 nF	Murata	GRM155R71C104KA88D	2	$0.10
TOTAL COST			$0.42		

4.1.8.4 Video Decoder

The analog video decoder requires 4 voltage supplies to function: Analog voltage supply AV_{DD} (3.3V), digital core supply voltage DV_{DD} (1.8V), PLL voltage supply PV_{DD} (1.8V), and digital I/O supply voltage DV_{DDIO} (3.3V).

The analog voltage supply AV_{DD} will be connected to the analog 3.3V power rail. Since both supplies provide voltage to analog systems and collectively draw less current than the regulator is rated for, they may be connected to the same power rail. Ultra-low ESR ceramic 10nF and 100nf decoupling capacitors will be used near the supply input pin of the video decoder to ensure the voltage supply is clean.

The digital core voltage supply and the PLL voltage supply will be connected to the digital 1.8V power rail. The regulator designated to deliver the 1.8V is rated for a maximum current draw of 3 amps. The current draw of two decoder supplies is estimated to be around 120mA according to the ADV7181 datasheet. This additional current draw will not hinder the operation of the 1.8V supply system. Ultra-low ESR ceramic capacitors will be placed near the appropriate supply pins on the video decoder to filter any noise from the supply voltage.

The digital I/O 3.3V supply will be connected to the 3.3V power rail designated for 3.3V digital supplies. Ultra-low ESR ceramic capacitors will be placed near the appropriate supply pins on the video decoder to filter additional noise. Table 32 shows the additional capacitors used for all of the video decoder power supplies.

Table 32: Required capacitors for AVDD, DVDD, PVDD, and DVDDIO supplies.

Supply	Value	Manufacturer	Part Number	QTY	Cost/Unit
AVDD	.1 µF	Murata	GRM155R71C104KA88D	1	$0.10
	10 nF	Murata	GRM033R70J103kA01D	1	$0.10
DVDD	.1 µF	Murata	GRM155R71C104KA88D	2	$0.10
	10 nF	Murata	GRM033R70J103kA01D	2	$0.10
PVDD	.1 µF	Murata	GRM155R71C104KA88D	1	$0.10
	10 nF	Murata	GRM033R70J103kA01D	1	$0.10
DVDDIO	.1 µF	Murata	GRM155R71C104KA88D	2	$0.10
	10 nF	Murata	GRM033R70J103kA01D	2	$0.10
TOTAL COST			$1.20		

4.1.8.5 Audio ADC

The AD1871 Stereo Audio ADC requires 3 voltage supplies to power its analog and digital systems. All three of these supplies require a voltage of 5 volts. The two digital supplies (digital core supply DV_{DD} and digital interface supply ODV_{DD}) can be taken from the digital 5V power rail while the analog supply AV_{DD} will be taken from the 5V rail designated for analog voltage supplies.

4.1.8.5. USB UART IC

The FT232 USB UART IC requires one supply voltage V_{CC} in the range of 3.3V to 5.25V to power the device core. V_{CC} can be supplied via the USB bus (when connected) or an external supply. Both the V_{CC} and V_{CCIO} pins will be connected to this supply and decoupling capacitor network. For the final design, the USB UART IC will receive power from an external supply; the 5V power rail designated for 5V digital supplies. Decoupling capacitors will be included to filter ripple voltage at the supply inputs. Table 33 lists these capacitors. Decoupling capacitors are connected between V_{CC} and ground and are low ESR ceramic capacitors.

Table 33: UART IC Additional Decoupling Capacitors.

Value	Manufacturer	Part number	QTY	Cost/unit
100 nF	Murata	GRM155R71C104KA88D	2	$0.10
4.7 µF	Murata	GRM188R60J475KE19D	1	$0.12
Total cost			$0.32	

4.1.8.7 Infrared Receiver

The Infrared receiver requires a 2.5V to 5.5V supply to operate. The TSOP75…W claims to be insensitive to the ripple voltage of the power supply so no additional filtering capacitors are needed on at the supply input pin. Since the IR receiver draws a maximum

current of 3mA, it can be connected to the 5V power supply designated for analog voltage supplies. The ground pin of the receiver will be connected to ground.

4.8.1.8 Shift Registers

The SN74HC165 parallel-load shift registers require an operating voltage range of 2 to 6 volts. Since the nominal supply voltage value given in the datasheet is 5V, these devices will be connected to the 5V power rail designated for digital supplies. Five parallel-load shift registers are needed in the Super-Doubler. Since the maximum supply current drawn by this device is 10µA, all five of these devices can be connected to the same 5V rail.

4.1.8.9 Video Buffer

The THS7353 Video Buffer requires a supply voltage in the range between 2.7V to 5V. This device will receive power from the 3.3V supply rail designated to power 3.3V digital devices. It is recommended that a .1µF to .01µF decoupling capacitor be placed very close to the supply input pin of the video buffer. These capacitors remove unwanted noise from the supply and placing these capacitors farther away increase the likelihood voltage ripple and oscillation. A 100µF decoupling capacitor placed on the power rail is also recommended to reduce low frequency noise. Table 34 displays information pertaining to these capacitors. All of the recommended capacitors will be used in the final design.

Table 34: Video Buffer Addition Decoupling Capacitors.

Value	Manufacturer	Part number	QTY	Cost/Unit
100 µF	Murata	GRM31CR60J107ME39K	1	$0.81
.1 µF	Murata	GRM155R71C104KA88D	1	$0.10
.01 µF	Murata	GRM033R70J103kA01D	1	$0.10
Total cost			$1.01	

All of these capacitors are ultra-low ESR ceramic capacitors.

4.1.8.10 Amplifiers

The LMH6645MF amplifiers require a supply voltage range of 2.5V to 12V to operate and up to 10 of these devices will be used in the Super-Doubler. The supply current for this amplifier is approximately .650mA at the typical operating voltage of 2.7V. All of the amplifiers in the system can be powered from the same 2.7V power rail.

4.1.8.11 Power Sequencer

For the system to function properly, a sequenced start-up is required for the FPGA and MCU. The LM3880 3-channel power sequencer will be used to control power-on and off for these devices. The outputs of the sequencer are flags that connect to the regulators'

enable inputs. The regulators will power-on when the flag goes “high” and power-off when the flag goes “low”. The sequencer sends the flags in a specific order allowing some power supplies to turn on before others. Table 35 shows the supplies and the order in which they must be sequenced as required by the system.

Table 35: Power Supply Sequence.

Supply	Sequence order
3.3V analog supply	1 (before the 2.5V supply)
1.0V digital supply	1 (before the 1.8V supply)
2.5V digital supply	2
1.8V digital supply	3

All other power supplies do not require sequencing. Since the 3.3V and 1.0V supplies can be powered in any order or simultaneously, one of them will not be sequenced with the sequencer. Instead, the supply will turn on when the entire system is turned on. The first flag of the sequencer will be connected to the 1.0V supply and will be the first of the sequenced power supplies to turn on. The second flag will be connected to the 2.5V supply which will be the second sequenced supply. The last flag of the LM3880 will be connected to the 1.8V supply and will be the final sequenced supply. All other supplies will power-on when the Super-Doubler is initially turned on.

During power-off, the sequencer will turn off the connected supplies in the opposite order of which they were powered-on. The sequencer requires an operating voltage in the range of 2.7V to 5.5V. This device will receive power from the 3.3V rail designated for digital supplies.

The power system must be tested prior to having the PCB manufactured. Certain parameters must be tested and monitored to ensure the power distribution system provides the proper voltage supplies and transient characteristics needed to successfully power the parts in the Super-Doubler. A digital multi-meter will be used to test voltage levels of the power rails with respect to ground and the oscilloscope will be used to measure the peak-to-peak ripple voltage to ensure it remains with in the appropriate window of 10% (peak-to-peak) of the nominal voltage value.

4.2 Software

4.2.1 FPGA

4.2.1.1 Logic

The core modules of the FPGA logic are shown in Figure 42. The heart of the FPGA logic is the Image Scaler which is responsible for altering the resolution of the video frames. The Video Input Capture, Digital Video Output Controller, and Analog Video Output Controller modules are responsible for interfacing the FPGA with the video processing pipeline. The Control & External Communication module receives and interprets command words, known as code words, from the microcontroller. The commands from the microcontroller are using to alter the function of the Image Scaler module. Finally the GUI Generator contains all of the information and logic needed to generate the on-screen user interface.

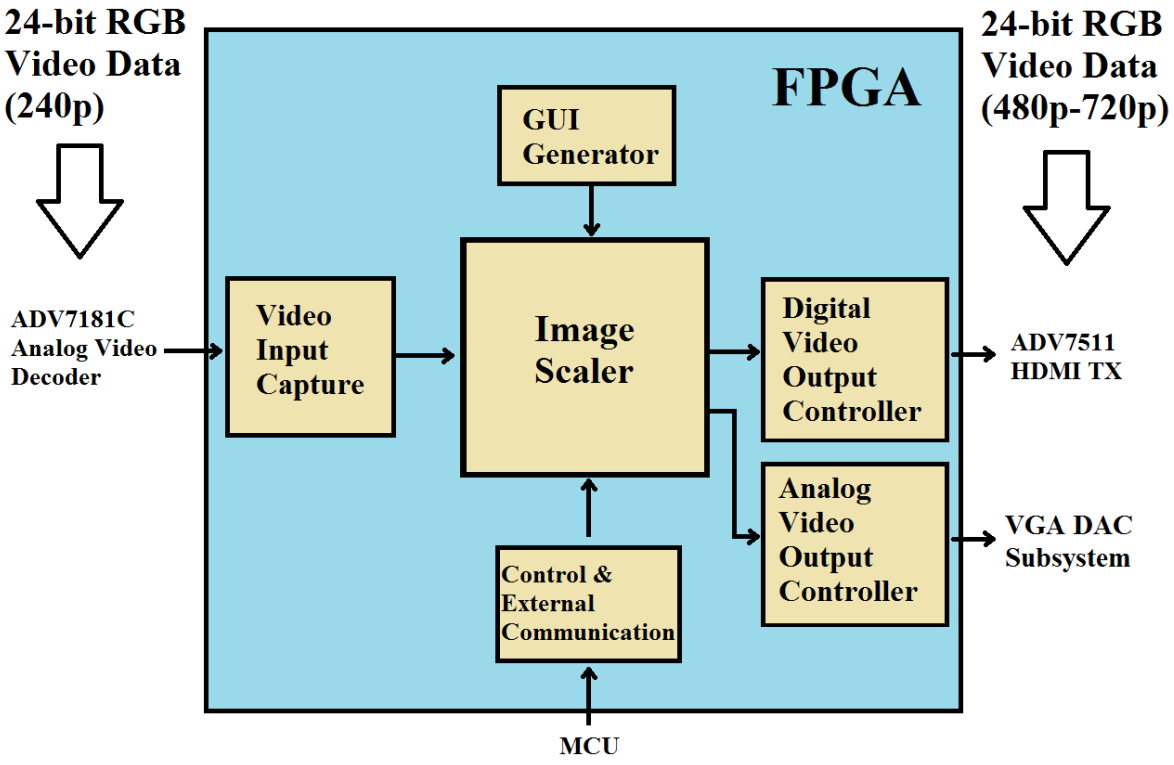


Figure 42: Top-level view of the FPGA system.

We will be implementing the above system using the Verilog HDL and Xilinx's Vivado Design Suite software. The XC7A35T from Xilinx's Artix-7 family of FPGAs will be used to realize this system. In the following sections we examine each of the main modules

individually with an emphasis on general data flow and behavior versus pinouts and schematics.

4.2.1.1.1 Image Scaler

The Super Doubler will support scaling by a factor of two and a factor of three without the use of a frame buffer. The lack of a frame buffer helps us achieve a low processing time per frame and will reduce the amount of input latency perceived by the user. Another benefit of forgoing the frame buffer is that our scaling module doesn't require external memory to store a frame. The Super Doubler will be able to scale an image using only three FIFO image row buffers. This results in a fast and area efficient scaling module. A functional block diagram of the Image Scaler module is shown in Figure 43.

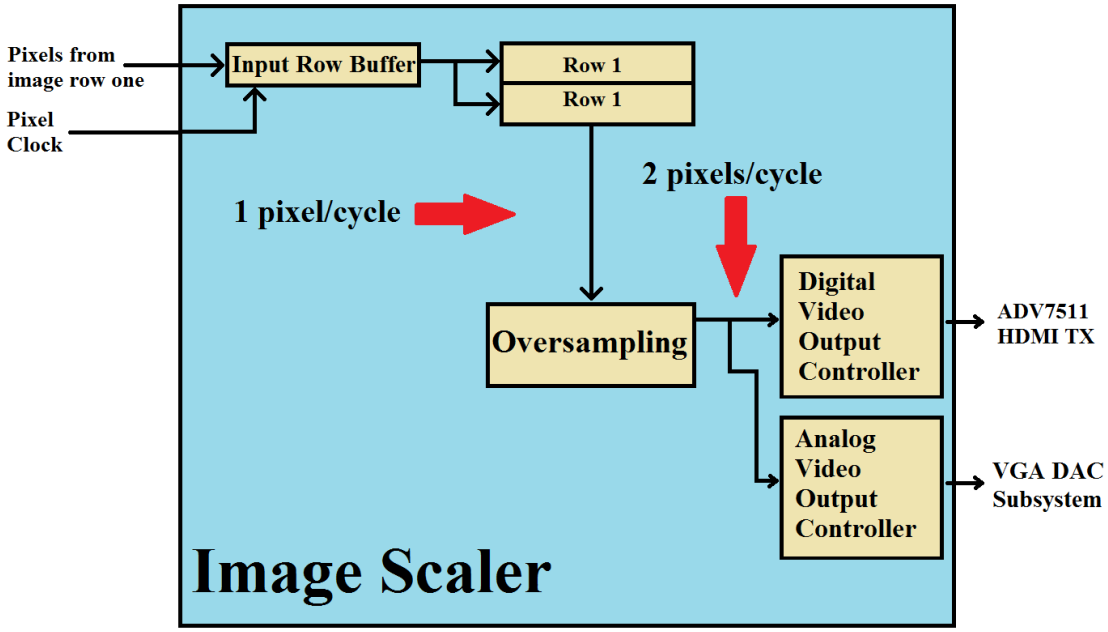


Figure 43: Functional block diagram of the Image Scaler module.

Scaling is accomplished in two phase: horizontally pixel-by-pixel and vertically row-by-row. A new pixel arrives at the Image Scaler input every pixel clock cycle. In order to scale the image horizontally we need to repeat some of the pixels in a row, a good method is to simply replicate every pixel twice. This is accomplished by oversampling the input pixel data at a rate two or three times that of the pixel clock. We use a phase-locked loop to generate the oversampling clock based upon the pixel clock. This process is shown in Figure 44 below.

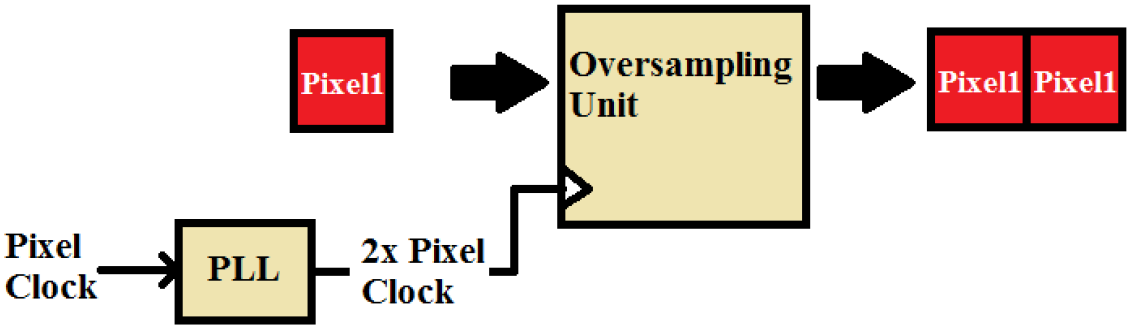


Figure 44: Pixel oversampling is used to achieve horizontal scaling.

Vertical scaling is accomplished by replicating every row of pixels two to three times. This is accomplished by storing the oversampled pixels in a FIFO row buffer as shown in Figure 43. When the row buffer is full, the contents are copied to a new row buffer that is twice as deep. This effectively gives us two repeating copies of an image row. The doubled row is then passed to the video output controllers where it is sent to the display. Combing the previously mentioned horizontal scaling method with this vertical scaling method, we can achieve scaling by an integer value. This method of scaling doesn't not use interpolation and is often called nearest-neighbor scaling. For video content with high contrast and sharp edges this method of scaling leads to an increase in brightness and size of the frame without softening the image.

4.2.1.2 Interfaces

4.2.1.2.1 Microcontroller Interface

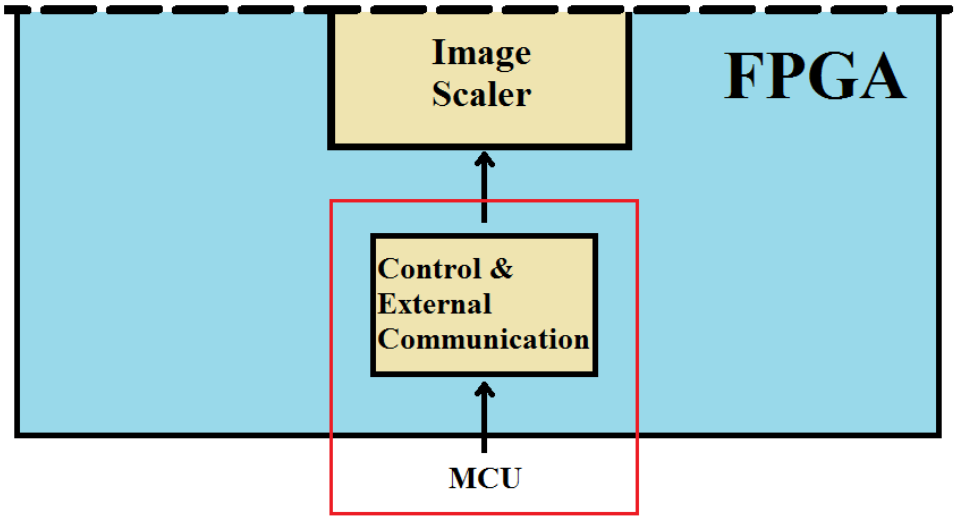


Figure 45: The subsection of the FPGA used for control and communication with the microcontroller.

Our system uses command words sent from the microcontroller to the FPGA to alert that FPGA that a change in operating mode is required. The microcontroller will send commands to the FPGA in the form of a 16-bit code word comprised of a 4-bit opcode and up to a 12-bit operand. Two external SN74HC165 shift registers will be used to pass the code word from the microcontroller to the FPGA in a parallel to serial fashion. The FPGA receives the control information from a single input port, therefore a simple finite state machine will be used to decode meaningful information from this serial stream. The overall responsibility of the finite state machine is to intake a serial stream whenever a flag from the microcontroller is asserted and output control information to the scaling data path. Tables 36 and 37 on the following page provides a summary of the code words and their related functions.

An STM Nucleo board will be used to test the physical interface between the FPGA and the microcontroller, this procedure is discussed in Section 7.1.4. After the physical interface has been validated we will begin testing the finite state machine which implements the control and external communication module. The control and external communication module provides the image scaler with a set of data which instructs the image scaler to alter its operation. To test this functionality we will implement a simple version of the image scaler module which processes a static video stream. We will then

provide code words to the image scaler in order to verify that it behaves correctly. The encoding format of the code words is shown in Table 36. The functionality of various opcodes are shown in Table 37.

Table 36: Encoding format of the full code word.

Full Code Word	
$C_{15}C_{14}C_{13}C_{12}C_{11}C_{10}C_9C_8C_7C_6C_5C_4C_3C_2C_1C_0$	
DATA = $C_{15}C_{14}C_{13}C_{12}C_{11}C_{10}C_9C_8C_7C_6C_5C_4$	**OPCODE** = $C_3C_2C_1C_0$

Table 37: Opcode functionality reference.

Function	OPCODE	DATA
Brightness Adjust	0001	Brightness Value
Sharpness Adjust	0010	Sharpness Value
Color Adjust	0011	Color select and value
Enable Dynamic Contrast	0100	Toggle. N/A
Enable Filter	0101	Toggle. N/A
Enable Scanlines	0110	Toggle. N/A
Scaling Factor	0111	Toggle. N/A
Not used	1001 to 1110	
Special Use	1111	

4.2.1.2.2 Audio/Video Pipeline Interfaces

The FPGA captures and transmits video data using three main modules as shown in Figure 46. The three modules are the video input capture module, the digital video output controller module, and the analog video output controller module. These modules are responsible for providing video data to the image scaler and properly timing the video outputs. The video input capture module interfaces with the ADV7181C analog video decoder. The digital video output controller provides signal timing and video data which

is used by the ADV7511 HDMI transmitter. The analog video output controller generates the timing signals used by the VGA output in addition to passing the 24-bit RGB video data to the output.

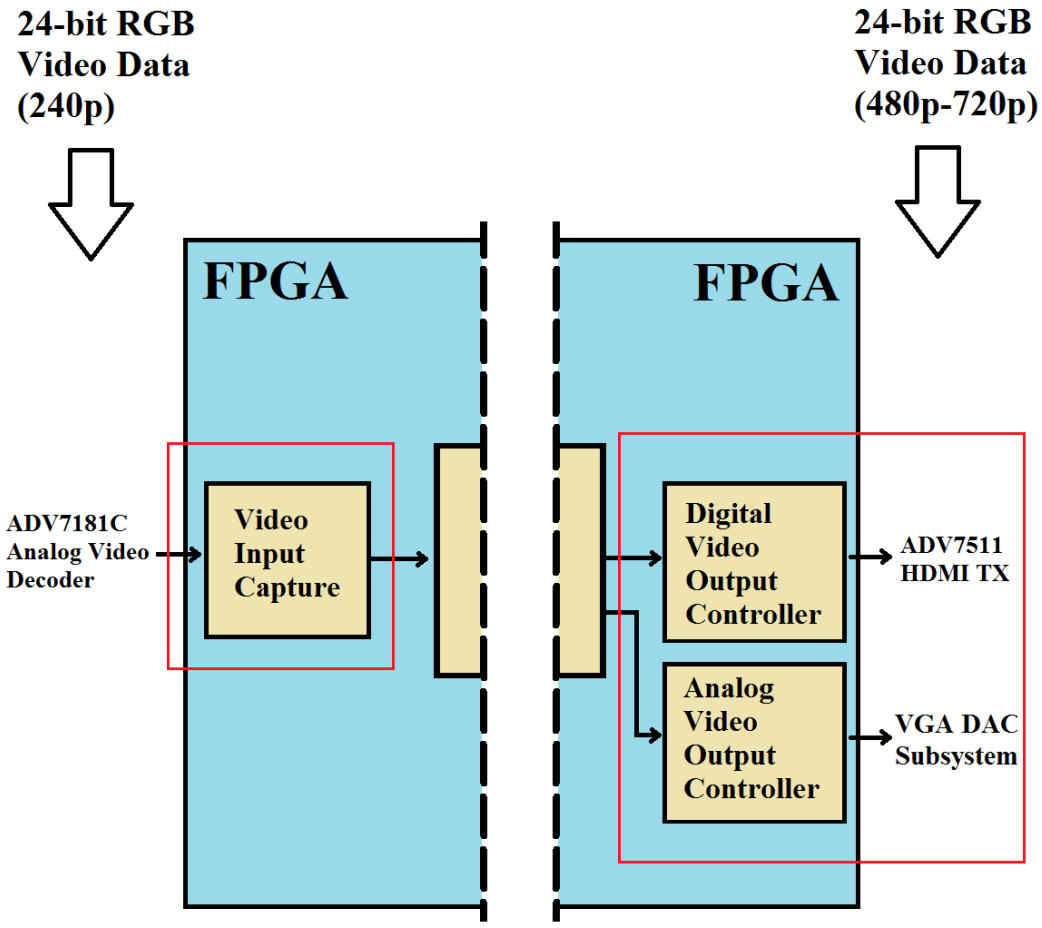


Figure 46: The subsections of the FPGA used for capturing input video data and formatting video at the output.

Testing will be done using the ARTY and KC705 FPGA development boards. The ARTY development board will be used to develop the analog video output controller and the KC705 development board will be used to test the digital video output controller. The ADV7511 is featured on the KC705 board which will let our testing procedure focus primarily on logic design without worrying about physical debugging. Figures 47, 48, and 49 show a lower-level perspective of the AV interface modules.

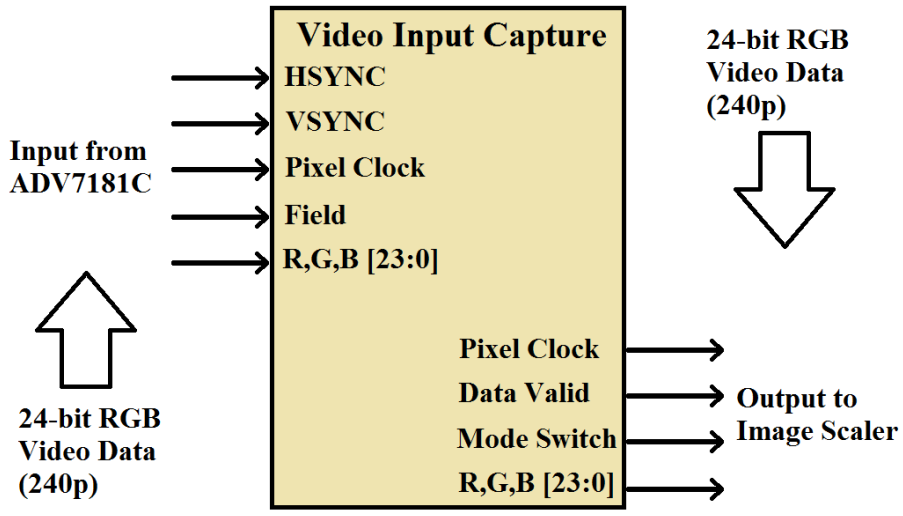


Figure 47: The Video Input Capture module receives input from the ADV7181C and outputs to the Image Scaler.

As shown in Figure 47, we see that the Video Input Capture module uses the timing information from the analog video decoder to generate RGB video data along with a data valid flag to alert the Image Scaler that the data can be processed. In Figure 48 below we see the Analog Video Controller module. The Analog Video Controller module receives pixel information from the Image Scaler module and outputs valid VGA video timing signals. The VGA timing signals are internally generated in the Analog Video Controller module based on the FPGA's master clock. A signal called Enable VGA allows the FPGA to shut down the VGA output if the user sets the device to digital output mode.

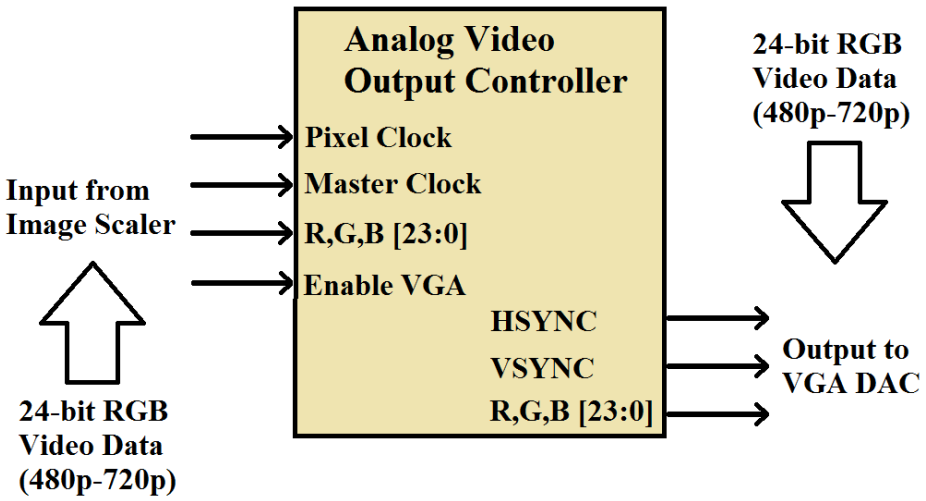


Figure 48: The Analog Video Output Controller module receives input from the Image Scaler and outputs to the VGA DAC.

The Digital Video Output Controller module is shown in Figure 49 below. The Digital Video Output Controller module is what drives the I/O ports connected to the ADV7511 HDMI transmitter. The Digital Video Output Controller module receives pixel information and timing from the Image Scaler module and uses that along with the FPGA's master clock to properly time the digital video output. A signal called Enable HDMI allows the FPGA to shut down the HDMI output if the user sets the device to analog output mode.

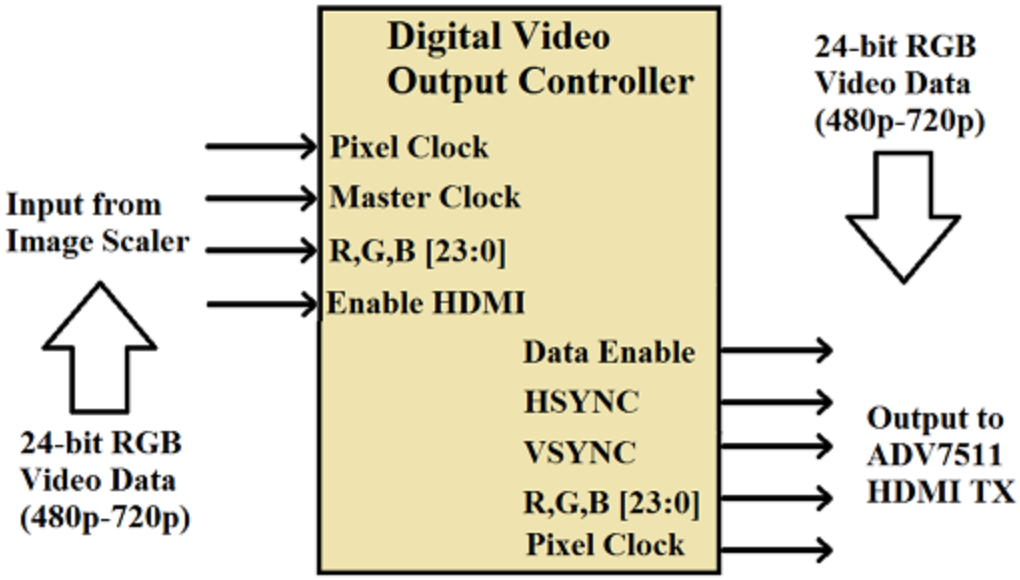


Figure 49: The Digital Video Output Controller module receives input from the Image Scaler and outputs to the ADV7511 HDMI transmitter.

4.1.2.3 GUI GENERATOR

The GUI Generator module uses control information from the microcontroller to inject on-screen display information into the Image Scaler module. The GUI is inserted at the Image Scaler module to ensure that it is properly scaled with video frame it will be displayed with. The GUI will consist of a single screen item display at a given moment that corresponds to what the user is manipulating. For example, if the user is adjusting the brightness then in the lower left of the screen <BRIGHTNESS> will be displayed, this is shown in Figure 50. If the user is increasing the parameter then the text on screen will show as green, if the user is decreasing the parameter then the text will show as red.

Figure 50. If the setting appears in green it means that the user is currently increasing the stored value of the setting.

4.2.2 Microcontroller

In this project the microcontroller takes on the role of initialization and coordination of the various discrete special purpose ICs in use. Much of this is startup configuration and acting as an intermediary between control devices and the components used to accomplish the video manipulation. The main responsibilities of the microcontroller are then programming and configuring the FPGA at power-on, accomplishing any startup configuration for the audio/video ICs, and then implementing the control and monitor role for the device in active operation. This section details the design of the various interfaces and processes required to accomplish these goals at the software level.

4.2.2.1 Logical Interface Description

The system view seen at the software level is very closely dictated by the communications requirements for each component. The STM32F070RBT6 was selected for this project particularly because of the large number of input/output busses and connections it supports, providing flexibility through the ability to change pin functions if more busses are needed and in selecting specific pin assignments. While the hardware connections used for the system are detailed elsewhere, the logical view of the components will remain decoupled from the hardware configuration.

I^2C is used extensively for both its protocol simplicity and small hardware requirement. Additionally, while the decision was made to use I^2C for the FPGA serial interface, both the ADV7181 video decoder and the ADV7511 HDMI transmitter only support I^2C for software configuration. Likewise with the SPI bus, the AD1871 audio decoder only supports software configuration via SPI. The SD card SPI interface was chosen as the STM32F070RBT6 does not provide dedicated SD card SDIO mode pins or software support as found in some higher end Cortex M3 and Cortex M4 devices. The SPI interface mode is used to overcome this by using the microcontroller's second SPI bus to avoid the added hardware and software complexity of multiplexing the SPI slave select pins.

In addition to I^2C and SPI, the STM32F070RBT6's USART1 bus is used to provide support for the vendor supplied bootloader. The STM bootloader detailed elsewhere in this document provides for reprogramming of the microcontroller via the chip's USART1 interface. This interface is exposed via the FT232RL USB-UART converter for USB reprogramming support, and most software for this purpose is preinstalled on the microcontroller itself. The Serial Wire Debug interface is also exposed for use of the STLINK/v2 debugging interface. Like the bootloader and USART1 bus, the software making use of this bus is outside of the developer view of the system.

For devices that do not require or support serial communication, the STM32F070RBT6's remaining GPIO capable pins are used for their control. This includes the pulse mode communications required for the IR sensor as well as control of the various LEDs used as on-enclosure user feedback. As there are a surplus of GPIOs available on the microcontroller and the FPGA only requires a rudimentary communications interface with

the MCU, free GPIOs are also used for FPGA communications features. At the software level, this consists of parallel communications not adhering to any outside protocol and special purpose flag pins used to provide signaling between the FPGA and microcontroller. The communications specification for MCU-FPGA communications is detailed in the section Section 4.2.1.2.1. Figure 51 provides a summary of the logical connections between the microcontroller and other devices in the project system.

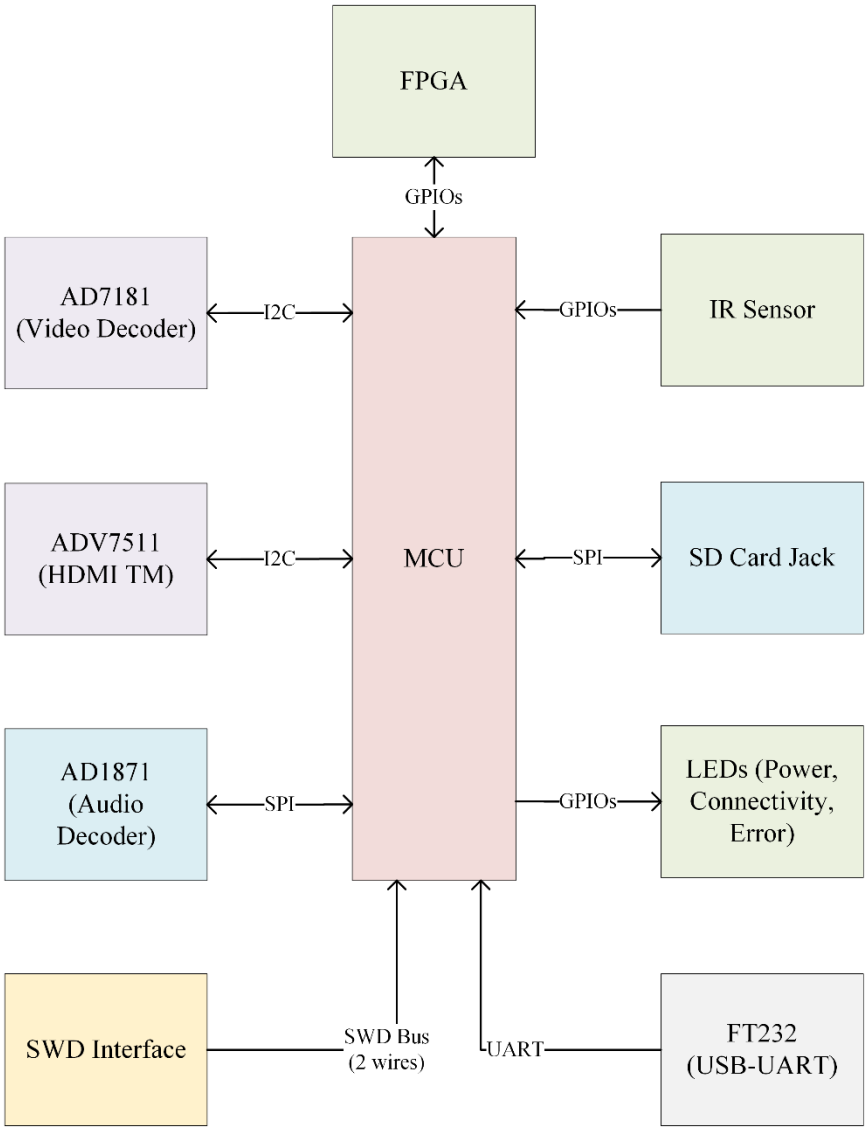


Figure 51: Logical view of microcontroller and discrete I/O device communications.

Access to each device will be accomplished via abstract access to its interfacing bus. For SPI connected devices this simply requires manipulation of the connecting slave select pin to signal device selection. I^2C enabled devices will all be connected via a single bus and accessed via their unique slave address. The logical slave addresses for I^2C devices in use in this system are shown in Table 38. Both audiovisual support ICs provide for configuration via a single pin to support two possible slave addresses for use in multichip

applications. The primary and secondary addresses are both shown for completeness and in the event one address is chosen over the other for implementation purposes.

Table 38: I^2C slave addresses for devices using the I^2C bus.

Device Name	Function	Primary Slave Address	Secondary Slave Address
ADV7181	Video Decoder	0x40	0x42
ADV7511	HDMI Transmitter	0x72	0x7A

4.2.2.2 Initialization Logic

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 configuring the FPGA from its accompanying binary file generated by FPGA development tools, 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. This section details initialization functionality outside of the FPGA programming, which is detailed in the In-System Programming section later in this document. As SD card use is intended primarily to support this feature, that interface is discussed there as well.

4.2.2.2.1 FPGA Post-Programming Initialization

After programming the FPGA, there will be a simple set of communications with the FPGA to verify correct programming and internal initialization before proceeding with the rest of the device startup procedure. A special data word will be set aside for use to instruct the FPGA over the MCU-FPGA communication link defined in Section 4.2.1.2.1 to raise a GPIO to indicate it is powered on, programmed, and properly interpreting communications. The sequence of events proposed is shown in Figure 52 for reference. The verification word communication and flag verification will be captured in a single routine VerifyFPGA(). After verification of the complete FPGA configuration and operation, the initialization can continue with audiovisual support ICs. If verification fails, the program will essentially terminate and loop indefinitely in a function IndicateFPGAFailure() to output a specific error sequence using the LEDs.

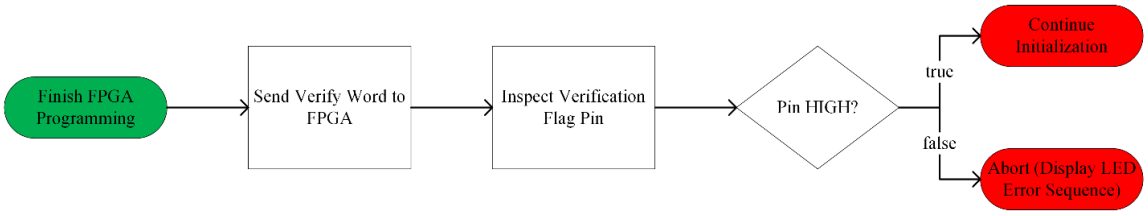


Figure 52: FPGA verification process sequence.

4.2.2.2.2 Special-Purpose IC Initialization

Initialization of the various audio/video support chips is required every time the device is powered on. This initialization is postponed until after the FPGA is fully configured as the initialization of these chips is both simpler and faster than the process required for the FPGA as well as the fact that they serve little purpose without FPGA functionality. Our design uses three chips from a single vendor, so the configuration processes are similar for each device. As the ADV7181 video decoder and ADV7511 HDMI transmitter both use the I^2C bus and the same register manipulation interface, the proposed configuration process for both are presented together. The configuration process for the AD1871 audio decoder is presented separately as it is different in both bus use (SPI rather than I^2C) and general process.

4.2.2.2.2.1 ADV7181 and ADV7511

The ADV7181 video decoder and ADV7511 HDMI transmitter must both be reconfigured at startup every time the device is reset before carrying out their video processing functions. As the chips are both made by the same vendor and likely designed to be used together in a single application, the configuration interface is almost identical between the two. For this reason, software level configuration of both is presented together here. Where differences apply they will be noted.

Each chip utilizes the I^2C bus to expose various internal memory mapped registers containing data words specifying configuration details. The microcontroller will have as part of initialization logic routines to send all necessary configuration words to each device. Upon entering this phase, the microcontroller will wait a short time of 200 ms as the ADV7511 requires a window at least this long between power on and being addressed over the I^2C bus. This window will be acknowledged even though FPGA setup likely required at least this much time simply to avoid difficult to find bugs due to not abiding by vendor device use guidelines. Writing to the various control registers uses the same procedural specification for both chips. Once the device is selected via the I^2C slave address, the next byte specifies the subaddress for a register. The byte following this is the data byte to be placed in that register. Figure 53 shows the sequence of these communications for reference for a data write.

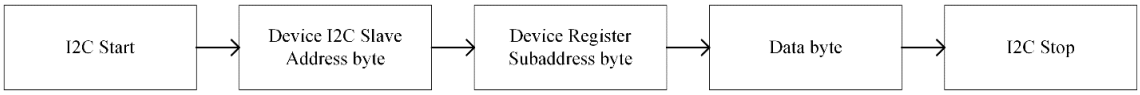


Figure 53: ADV device I^2C write sequence.

Data reads are accomplished by a similar process by writing slave address, followed by data register subaddress, followed by a duplicate I^2C start bit, followed by the device slave address again. The device then responds with the data in the register specified. A flow diagram showing the process is shown in Figure 54. As these processes are identical, two routines ADVWrite() and ADVRead() can be developed to interface with both devices. By

passing a parameter to the routines we can select a device to communicate with, but using the same process.

Figure 54: ADV device I^2C read sequence.

The ADV7181 can be configured as soon as the 200 ms window passes, essentially at device startup. There are over 150 registers to control the chip's operation, most of which will have to be determined in a later phase of the project through testing. Some of these registers support basic operation features such as video input signal specification and control of the formatting of the digital output signals. The ADV7511 transmitter is powered up and down over the device's operation to support HDMI hot plugging. Many registers, including the register specifying the status of the Hot Plug Detect line are addressable even when the device is powered down. This register can be polled repeatedly during operation to determine when to reinitialize the ADV7511. Reinitialization requires configuring of some registers that are cleared when the Hot Plug Detect line goes low (connection with external device terminated) and will be fit into discussion of the microcontroller's post-initialization operation in a later section. Figure 55 shows the general initialization of the ADV7181 and ADV7511 at initial device power on before proceeding into initialization of other parts in the system. Note that "Volatile" refers to configuration registers that are reset when Hot Plug Detect goes low and "Preserved" refers to registers that are not reset when Hot Plug Detect goes low.

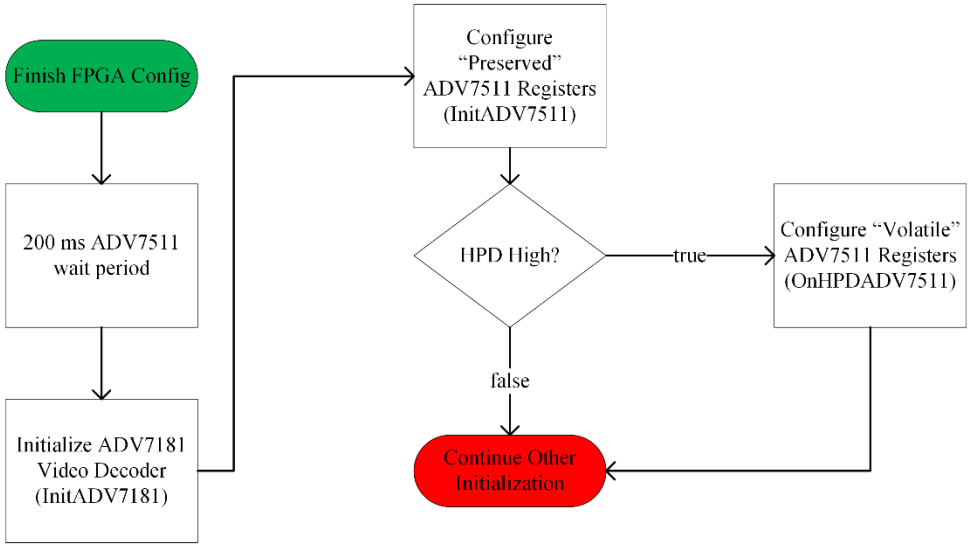


Figure 55: Video device initialization sequence.

4.2.2.2.2.2 AD1871

The Analog Device AD1871 is the audio decoder and digitizer being used for audio conversion in this project. Interfacing with the AD1871 is similar in concept to the process used for the ADV7181 and ADV7511 in the previous section, as it has a set of registers used for control of and feedback from the device. While the ADV7181 and ADV7511 use the I^2C bus for communications, the AD1871 uses a SPI interface. One of the two SPI interfaces on the STM32 microcontroller is dedicated to this interface. The AD1871 requires significantly less configuration to use, sporting just 5 addressable 10-bit registers, 2 of which are read only for audio information. Communication is initiated by pulling the device's slave select line low. The first 4 bits sent are the address of the register desired to be configured or read, followed by a read/write bit, a reserved bit, and then the 10 bit register data. If writing, these 10 bits are written by the MCU to the device, otherwise these 10 bits are written from the device to the MCU. Figure 56 shows a diagram of this bit sequence. Note that address and data bits are written MSB first.

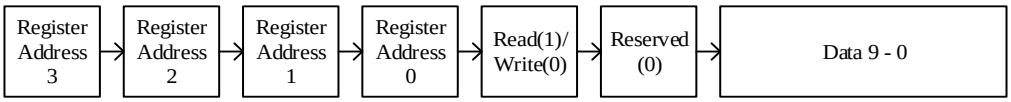


Figure 56: AD1871 SPI configuration bit sequence.

All configuration of the AD1871 is expected to be done at initialization. While the control registers do support toggling functionality like mute and input data sources, these settings are not needed to be modified for our purposes. This will necessitate a single routine to be called after video device initialization, tentatively called InitAD1871().

4.2.2.2.3 IR Interface Timer Clock Initialization

STMicroelectronics makes available a set of example programs and libraries for use with STM32F0xx evaluation boards. These demo programs include libraries for using the advanced Timer modules in the microcontroller for interrupt based monitoring of the IR LED input pin for capturing IR commands. Both initialization and decoding routines are provided for RC5 and SIRC IR protocols, and ST provides documentation with guidelines for modifying the libraries for use with any IR protocol. These libraries will be used and modified as necessary to provide IR interface capabilities for this project. Part of the libraries' use requires initializing the timer clock and interrupt registers for the timer module used in implementation. These routines are referred to as IR_Init() where IR is replaced with the protocol used (e.g. RC5_Init()). This initialization is shown relative to preceding and subsequent program execution for quick reference in Figure 57.

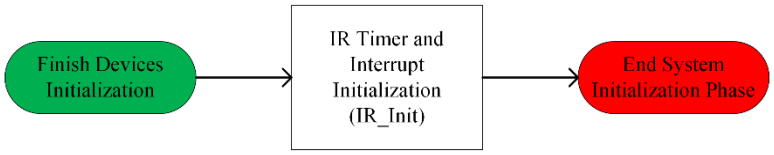


Figure 57: IR timer and interrupt initialization and surrounding processes.

4.2.2.2.4 Initialization Routines Summary

The processes outline in this section dictate that several high level routines exist for carrying out the necessary tasks. Those routines are mentioned in the preceding sections where relevant and provided here in Table 39 for quick reference of name and intention.

Table 39: High level initialization function reference.

Function Name	Description
VerifyFPGA()	Send the verification word to the FPGA and verify that it is programmed and interpreting commands properly.
IndicateFPGAFailure()	Blocking/looping function to indicate to the user that FPGA configuration failed or MCU to FPGA communication has failed
ADVWrite()	Write to one of the control registers on the ADV devices. Device, register address, and data byte are all to be specified as parameters to make this function device independent
ADVRead()	Read from one of the control registers on the ADV devices. Device and register address are to be specified as parameters to make this function device independent
InitADV7181()	Perform initialization via control registers of the ADV7181 video decoder
InitADV7511()	Perform initialization via control registers of any registers in the ADV7511 HDMI transmitter that are not reset on HPD low
InitAD1871()	Perform initialization via control registers of the AD1871 audio decoder
OnHPDADV7511()	Initialize control registers that are reset when HPD low causes the ADV7511 to power down. This reinitializes the HDMI connection and is used throughout the program as HPD is polled as HIGH and the device is powered on
IR_Init()	Initialize internal STM32 functionality for handling IR input. Reference code provided by ST libraries

4.2.2.3 Control Logic

After initializing all of the special purpose ICs and capabilities of the microcontroller, the microcontroller logic will enter an indefinite control loop. This loop is intended to facilitate

system control and feedback between the user and the device. This control functionality is detailed in this section.

The system provides two means of user feedback and control, the IR remote control and the on-enclosure LEDs. The remote control allows the user to adjust various runtime configuration parameters such as controlling video processing features and navigating the user interface, while the LEDs are intended to provide basic feedback about system operation. The microcontroller handles the entirety of the processing related to this functionality. This functionality will be implemented as a single loop iterating indefinitely and capturing all necessary user interactions and device status information updates.

4.2.2.3.1 IR Decoding

The user control requests via the infrared remote are interpreted by the microcontroller as part of the main control loop and decoded into meaningful commands for the FPGA. This decoding will be accomplished using a modified version of the example library provided by STMicroelectronics for STM32F0 evaluation boards. The library functions essentially poll for IR pulse data at regular intervals defined by a timer based interrupt. Using the PWM input mode of the given timer interrupt, register flags are raised and times are recorded on rising and falling edges of the pulse. The time between these flags is then used to determine the pulse characteristics. Interrupts are timed such that they occur once during every pulse and pulse values are recorded one bit at a time. If the pulse cannot be reconstructed, the pulse bit string is reset. When a valid complete pulse string is successfully received, a flag is raised to signal that a complete IR frame is ready to be decoded when required in the main control loop. This frame structure contains all of the necessary IR transmission request data including the sending device and button pressed. A high level depiction of this process is presented in Figure 58 for reference.

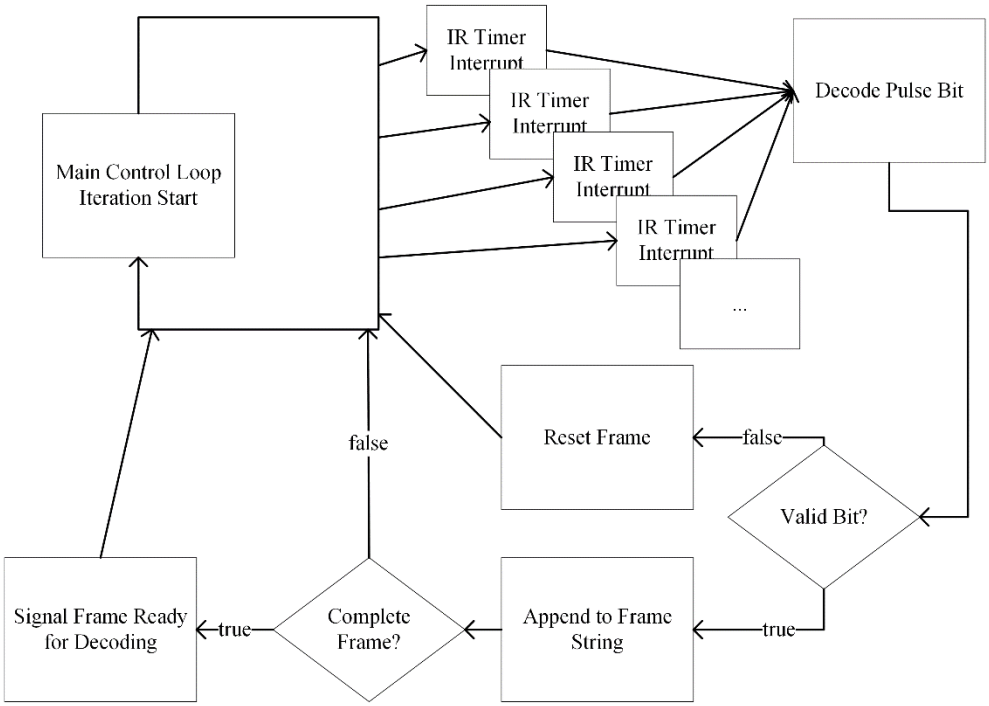


Figure 58: IR pulse decoding logic flow.

IR frames are decoded to determine the button pressed by the user and the appropriate action to take. This is then coded into the command word sent to the FPGA according to the protocol established in the section detailing MCU and FPGA intercommunications via the ParseIRFPGACommand() function.

4.2.2.3.2 FPGA Verification and Signaling

Commands to be passed to the FPGA are decoded from the IR commands received. Following successful command decoding, the command will be sent to the FPGA according to the protocol previously established for processing via parallel loading of the interfacing shift registers via the SendFPGA() function. The FPGA will be signaled using a dedicated flag line as part of this function that a new command is available for processing. In addition to this command, the FPGA communications will be periodically verified using the same routine defined in the initialization design section. This verification is intended to help detect and diagnose problems in the FPGA MCU communications interface that appear after FPGA programming and initialization. The results of this will be tracked and used in determining whether to follow IR command decoding with transmission to the FPGA interface and to determine the output of the FPGA status LED. Complete device failure will not be forced as the FPGA may still be functioning internally, but the developer and user must be notified that IR communications will be nonfunctional at this time. Verification will be reattempted regularly to prevent intermittent errors from resulting in complete system halts and allow for FPGA and MCU reconnection. A graphical overview of the FPGA communications procedures used in the main control loop is provided in Figure 59.

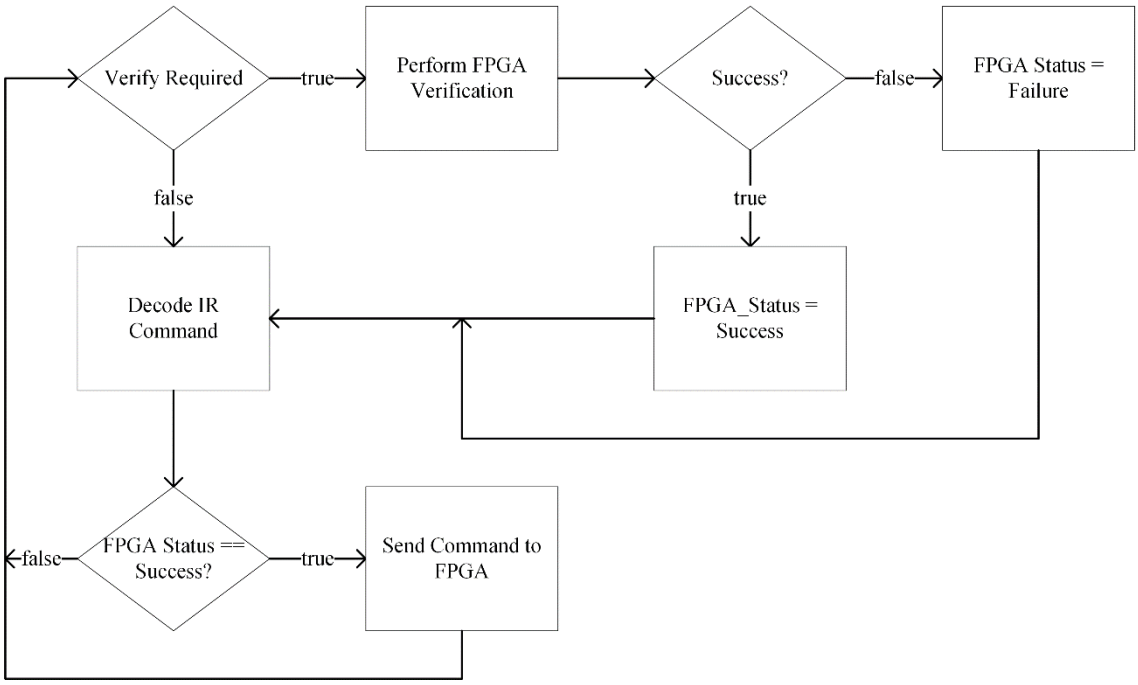


Figure 59: FPGA verification and MCU communication logic flow.

4.2.2.3.3 HDMI Transmitter Control

The ADV7511 HDMI transmitter must be continuously monitored for the Hot Plug Detect feature of the HDMI standard indicating that the HDMI connector has been connected to an external device. The transmitter must then be powered on and reinitialized as discussed in the initialization logic section concerning the device. At power on, all of the control registers lost on power down must be reconfigured to reestablish the HDMI data link with the external device.

This will be accomplished via simply polling the register containing the HPD line status in the ADV7511. If HPD is low, signaling that there is no connection, the device will be powered down if it not already using the OffHPDADV7511() function, resulting in loss of HDMI connection configuration register data. If HPD is high, the device will be powered on if not already and reinitialized using the same OnHPDADV7511() function discussed in the initialization section.

4.2.2.3.4 LED Control

The on-device LEDs will be used to provide the user with feedback concerning the status of the system. They will be updated every control loop iteration according to continuously tracked device status concerning FPGA communication status and HPD status. FPGA communication status informs the user whether a serious error has occurred between processing components and HPD status informs the user whether the MCU is aware of

output device connections necessary to power on the HDMI transmitter. After initial configuration, the LEDs will be controlled simply as either on or off depending on device status. On will correspond to successful FPGA communication for that LED and HPD high for the HPD led. The GPIO states used to control them will simply be updated every iteration via an UpdateLED() function.

The LED subsystem is also used to indicate failure at various phases of system initialization in indefinite error sequence routines. These LED sequences are described in the test plan section and the routines referenced in appropriate sections where the failure may occur.

4.2.2.3.5 Control Routines Summary

The processes outline in this section dictate that several high level routines exist for carrying out the necessary tasks. Those routines are mentioned in the preceding sections where relevant and provided here in Table 40 for quick reference of name and intention.

Table 40: High level control function summary.

Function Name	Description
IR_Decode()	If a complete IR frame has been received, decode it into the structure specified via the protocol in use. Reference code provided by ST libraries.
ParseIRFPGACommand()	Parses the command structure received from IR_Decode() into the appropriate command word to send to the FPGA
VerifyFPGA()	Send the verification word to the FPGA and verify that it is interpreting commands properly.
SendFPGA()	Send a command word to the FPGA
ReceiveFPGA()	Receive a requested word from the FPGA
PollHPD()	Retrieve the register word containing HPD information from the ADV7511 and determine if HPD is high or low
OnHPDADV7511()	Initialize control registers that are reset when HPD low causes the ADV7511 to power down. This reinitializes the HDMI connection and is used throughout the program as HPD is polled as HIGH and the device is powered on
OffHPDADV7511()	Power down and reset the ADV7511 when HPD is detected low
UpdateLED()	Update LEDs status according to FPGA communications status and HPD line detection

4.2.2.3.6 Summary

This section detailed both the initialization and control design necessary for typical expected device operation for the various subsystems used. A summary of the complete MCU logical operation can be found in Section 5.2 of the Software Design Summary

4.2.3 In-System Programming

Due to integration of ICs with packages containing large pin counts and FPGA volatility, pre-programming via program flash devices and replacing the parts on the PCB is not a practical solution for this project. Additionally, reprogramming capability is desirable for both developers looking to test changes quickly and knowledgeable end-users looking to customize their devices. For these reasons, complete reprogramming capability was deemed necessary for this project.

4.2.3.1 Microcontroller Reprogramming

The ST microcontroller used in this project will be supported with two means of reprogramming. First, and intended for development purposes, the ARM Serial Wire Debug interface will be supported, allowing flashing and debugging from within IDEs such as Keil MDK and other support software used for development on the microcontroller. This interface allows for more interactive development methodology, but requires extra equipment not expected to be owned by the end-user. For this reason, a USB interface is also provided for easy firmware updates of the microcontroller software. The STM32 microcontrollers have a factory loaded bootloader capable of flashing the microcontroller program memory with provided binaries using the USART1 channel. A software application called ST Flash Loader Demonstrator is also provided by ST that implements this programmability. As both of these solutions are provided by the chip vendor and no software must be written for this project to support them, their implementation is outside the scope of this document and we must only provide the hardware interfaces necessary.

4.2.3.2 FPGA Reprogramming

FPGAs must be programmed every time they are reset. Vendors typically provide many ways to accomplish this. In an effort to make our system as flexible as possible, the microcontroller will program the FPGA as part of its initialization routines. The binary file used to program the FPGA will be supplied by the user via an SD card, whereby the file will be loaded and transferred to the FPGA according to the FPGA programming interface. This section details this process. As the SD card interface is provided explicitly for this programming capability, it is discussed in this section as well.

4.2.3.2.1 SD Card Interface

The logical and filesystem SD card access interface will be provided mostly by the FatFs library. It provides the entire filesystem implementation for handling SD card devices formatted in one of the FAT filesystem implementations, requiring the library user to only provide the implementation for interacting with and controlling the storage device. The storage device control functions required by the FatFs library to be implemented by the library user are summarized in Table 41 below.

Table 41: FatFs device control interface layer.

Function	Description
disk_status()	Get the status of the target device. Used to determine whether disk is initialized or write protected.
disk_initialize()	Prepares the target device to be ready to accept read/write requests
disk_read()	Read a specified number of disk sectors beginning at a supplied target sector into a buffer
disk_write()	Writes the data provided in a supplied buffer to a specified set of sectors on the disk
disk_ioctl()	Used to control and query device parameters such as sector size, block size, and device status information
get_fattime()	Get the real time clock reading with the year relative to a start point of 1980. Used for timestamping files. We do not expect to perform filesystem writes so this function is not critical.

These functions will be implemented with help from the STM32 Standard Peripheral Library support for the MCU SPI interface to provide the complete SD card interface in SPI mode. The STM32CubeF0 software package also provides sample disk interface functionality for FatFs as required. This source can be modified and used to provide the device control layer necessary for our implementation without the need for writing the hardware driver from scratch.

4.2.3.2.2 FPGA Slave Serial Configuration

The Xilinx 7-series FPGAs support startup configuration via a variety of means. To conserve on pins needed and to accomplish configuration entirely via the microcontroller and SD card in the system, the slave serial configuration mode will be used. In this configuration mode, the FPGA binary is loaded in to the FPGA one bit at a time, verified, and the programming device (in this case the MCU) notified of success.

To accomplish this configuration, a binary configuration file will be provided by the developer or user on the SD card inserted into the SD card jack. At system reset, the MCU will check for a variety of filenames determined throughout project development and open the first match it finds via the SD filesystem interface. If a valid filename cannot be found, initialization will fail with an LED error code. A 32 bit word will be loaded from this file, and shifted one bit at a time out to the FPGA serial programming interface. This process will repeat until the entire file has been loaded into the FPGA. The FPGA serial programming interface then defines a wire that will be driven high by the FPGA to indicate programming success. After this programming has been determined successful, the microcontroller will continue to verify its communication path with the FPGA and move into the rest of the system initialization. If at any point failure is determined, the microcontroller application will move to an error sequence to display an error pattern indefinitely on the LEDs to indicate FPGA programming failure. A graphical summary of this configuration process is shown in Figure 60.

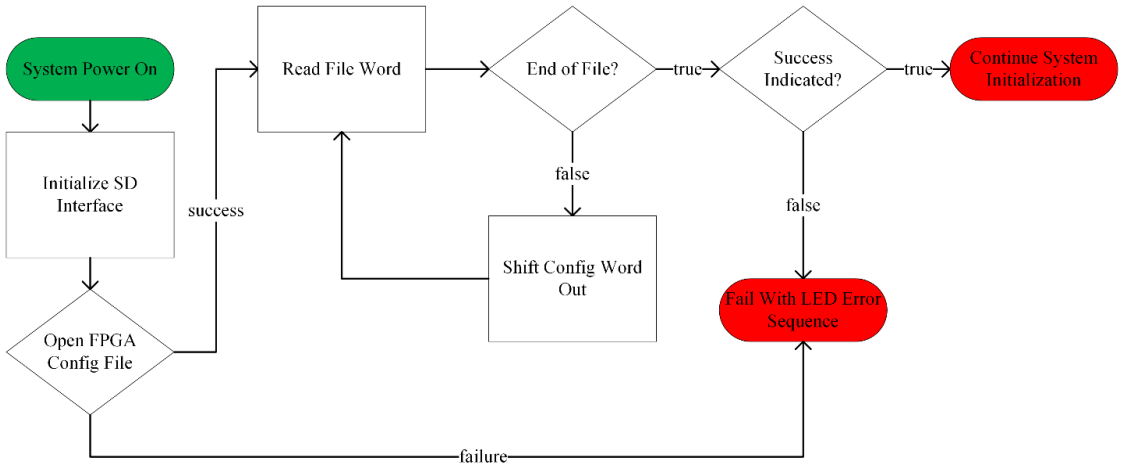


Figure 60: FPGA programming configuration flowchart.

This process will require a number of high level functions for implementation. Those readily identified are summarized in Table 42 with brief descriptions of their purpose. Note that SD card filesystem initialization and manipulation is accomplished via FatFs library functions.

Table 42: FPGA configuration high level function summary.

Function	Description
ReadFPGAConfigWord()	Read a word from the FPGA configuration binary. Must also indicate when the end of the file has been reached
ShiftFPGAConfigWord()	Shift a read FPGA configuration word out to the FPGA over the serial programming interface one bit at a time
CheckFPGASuccess()	Check for FPGA programming success indication after loading the entire configuration file
FPGAFileNotFound()	Failure routine to indicate indefinitely on the LED output that an appropriate FPGA configuration binary was not found
FPGAProgramFailure()	Failure routine to indicate indefinitely on the LED output that programming of the FPGA from the configuration binary failed

5. Design Summary

5.1 Hardware Design Summary

5.1.1 System Block Diagram

A complete system block diagram is shown in Figure 61 on the following page. This diagram summarizes the devices used in the system and the data passed between these devices during complete system operation.

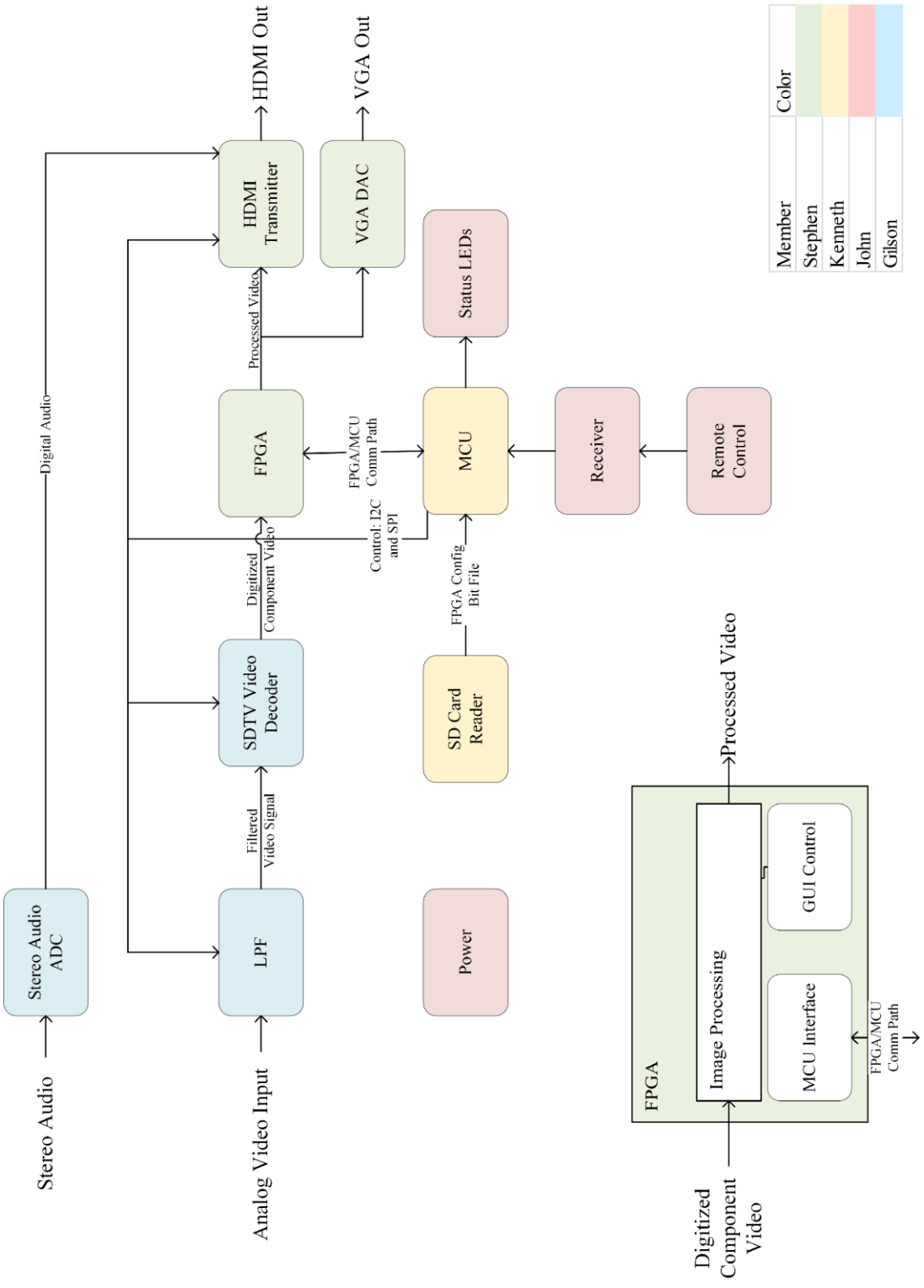


Figure 61: System block diagram overview.

5.1.2 Microcontroller Interface Connections

Microcontroller connections to the other ICs in the system require specific pin assignments due to GPIO alternate functionality and bus assignment via the microcontroller. The connections mandated for the desired functionality with the microcontroller and other IC packages are detailed in Figure 62. Note that connections between the MCU and FPGA as well as LEDs are not detailed here. These are all to be connected to the MCU via the MCU's free GPIOs and the specific pin assignments will be determined later as board layout requires during PCB design. The MCU pins used are 5V tolerant where required, so level shifters were not required for any device interfaces.

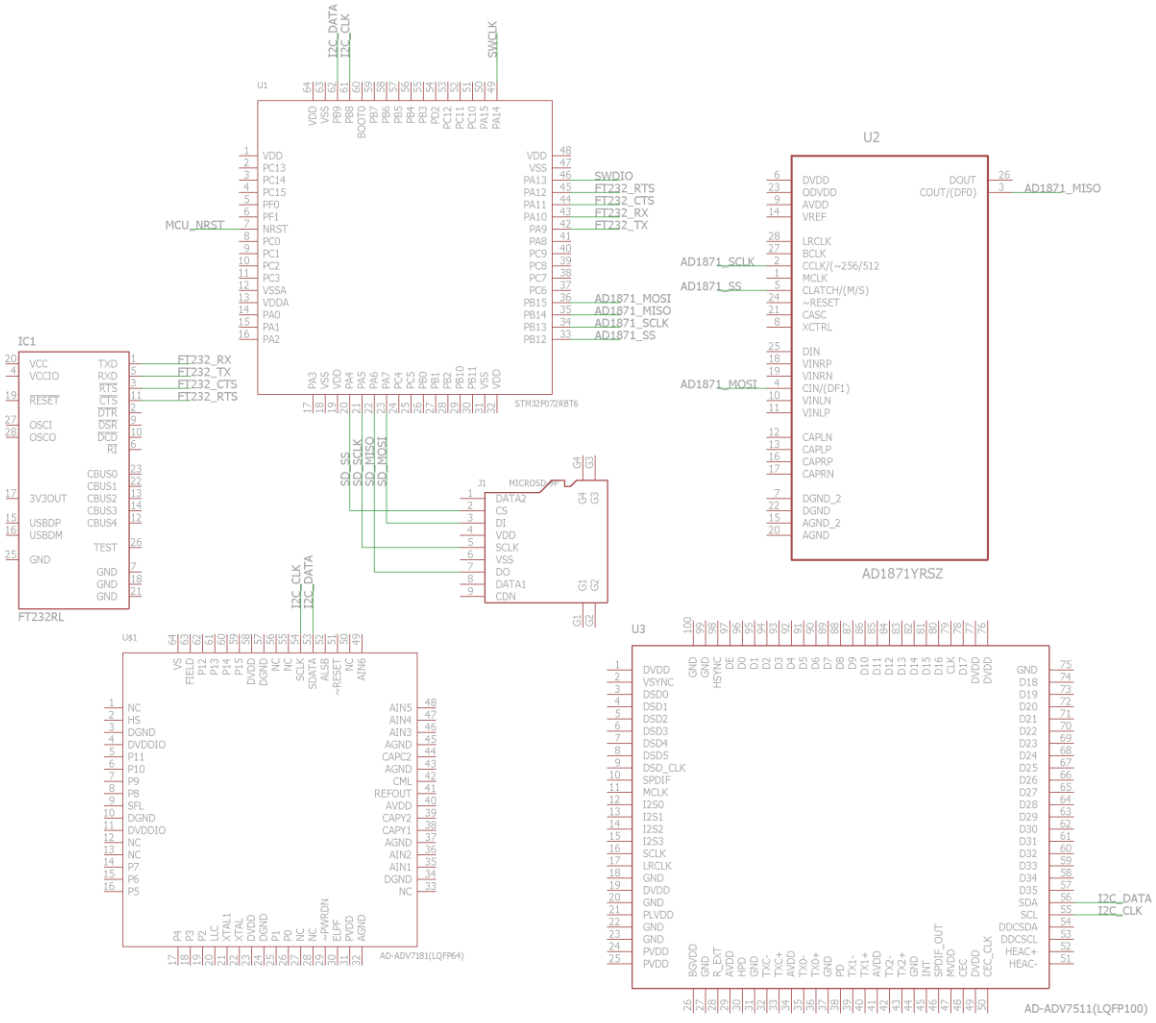


Figure 62: Microcontroller pin connections to system ICs.

5.1.3 PCB Mockup

Shown in Figure 63 below is a planned layout of the major components on the printed circuit board. The audio and video inputs are arranged on the left side of the board with the audio and video decoders close by. This portion of the PCB design will require careful layout as the analog signals which traverse it are very prone to signal loss and noise sensitivity. The video output ports will be opposite of the portion of the PCB containing the user interface material such as status LEDs and the SD card.

Components related to power and other auxiliary tasks are not shown in an effort to emphasize the major signal paths that will need to be routed during PCB layout.

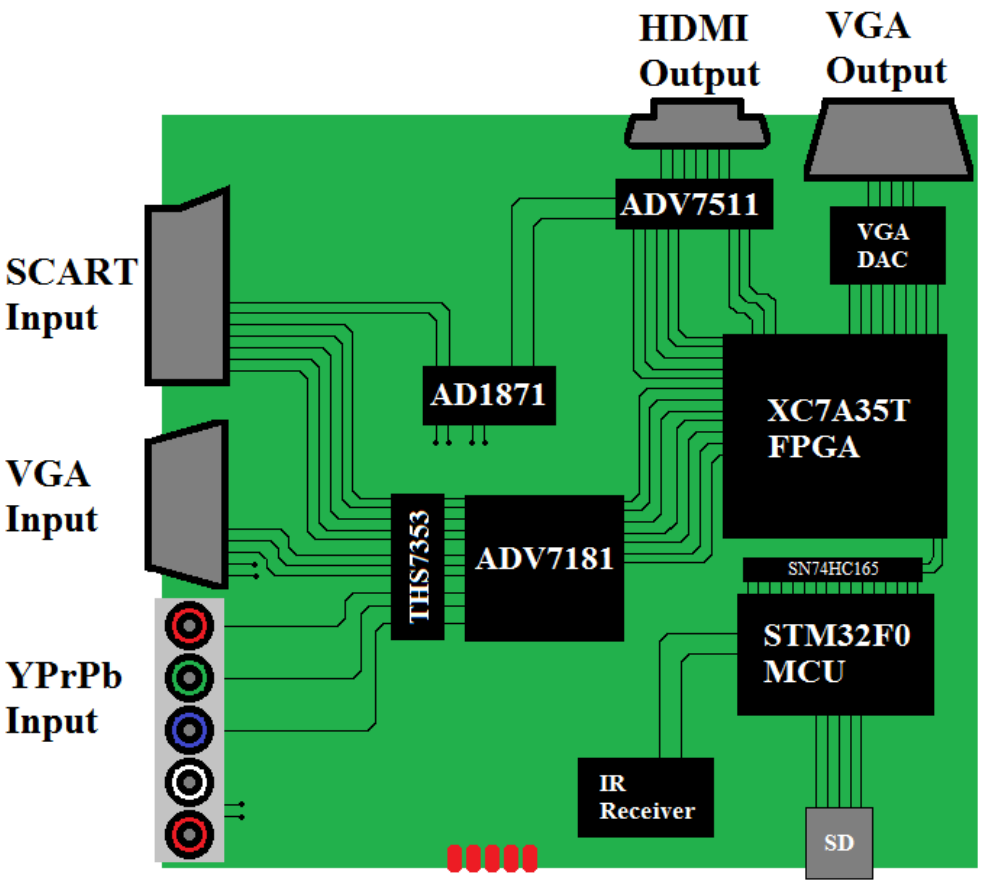


Figure 63: A mockup of the PCB layout showing the major components in the signal path.

5.2 Software Block Diagrams and Summary

5.2.1 FPGA Logic Summary

An overview of the main logic paths in the FPGA is shown in Figure 64.

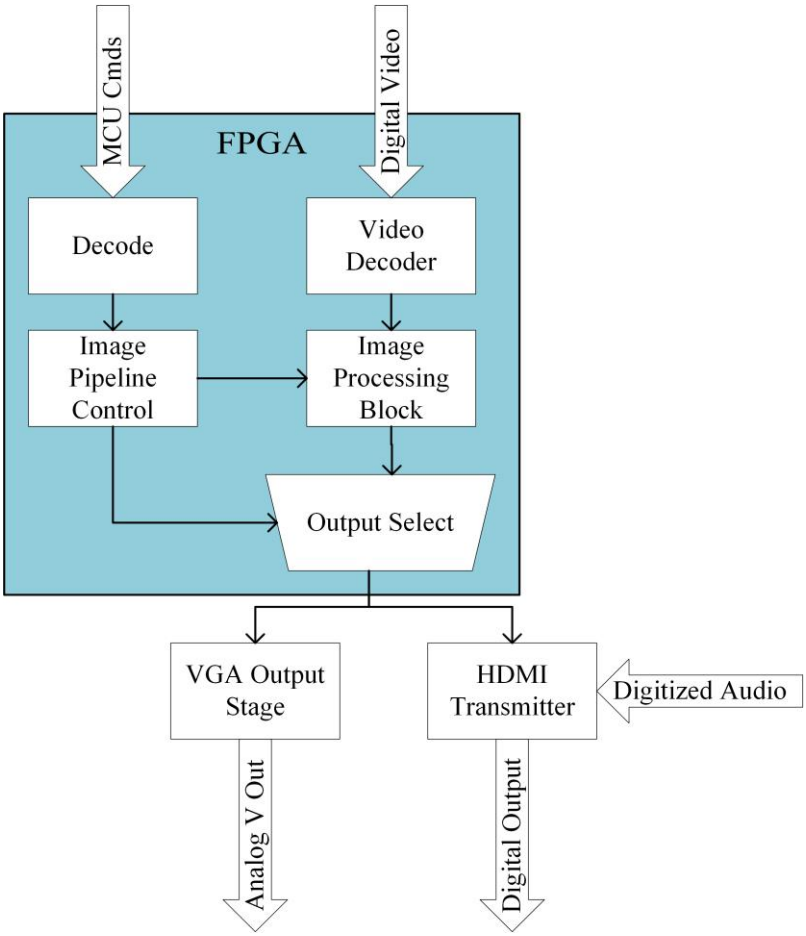


Figure 64: FPGA logic overview.

5.2.2 Microcontroller Logic and Flowcharts

5.2.2.1 Initialization

A complete summary of the microcontroller initialization processes between system reset and entry into the control phase of the application is shown in Figure 65. This phase of the program initializes every subsystem and verifies that developer controlled applications such as the FPGA and SD interface are correctly functioning. On failure the application indicates the failure using specific LED sequences and codes for easy diagnosis of where

the problem is likely to be found. These error codes can be found in the test plan for the microcontroller interfaces.

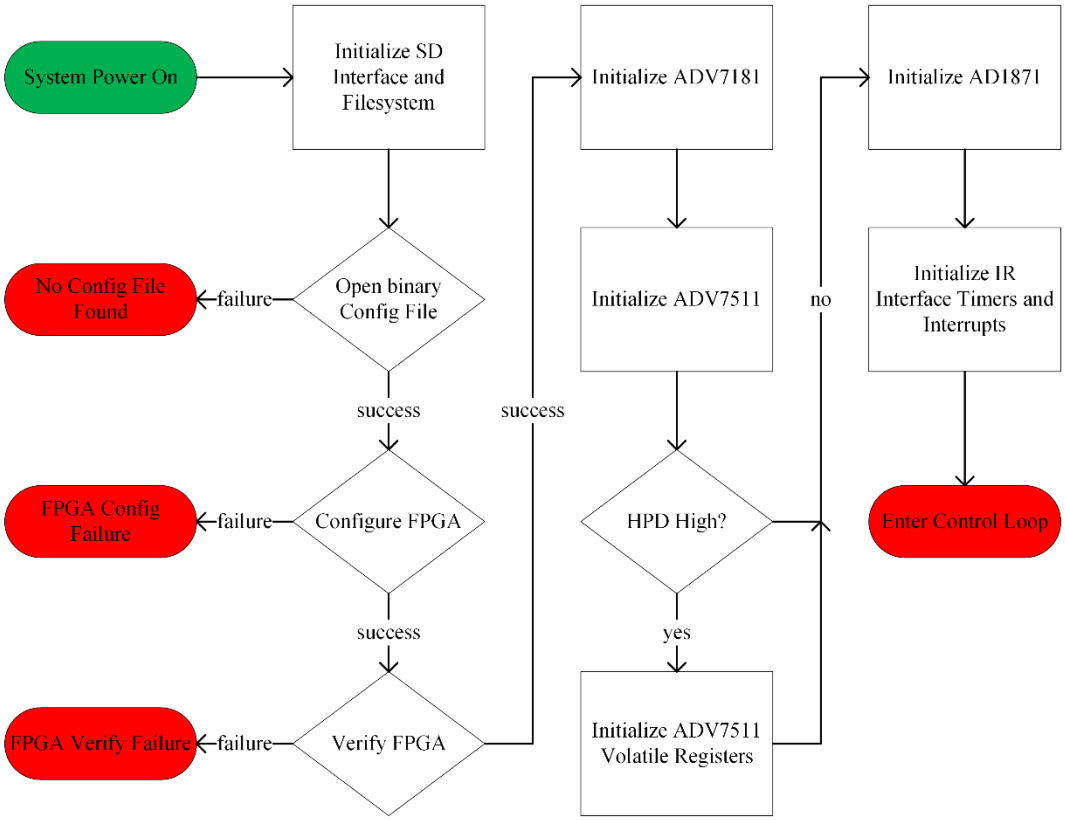


Figure 65: Microcontroller initialization processes summary.

5.2.2.2 Control

A complete summary of the control logic used after system initialization for the microcontroller is shown in Figure 66. In this control loop, the program is essentially acting simply to provide a control interface for the user via the infrared remote control and feedback about detection of the system ICs via the LEDs. Checks like FPGA verification step and HPD detection also update the current states of the device necessary in the LED update at the end of the loop. Additionally, each HPD result must first check the power status of the device to determine if the operation needs to be performed. This control loop runes indefinitely until the device is powered down.

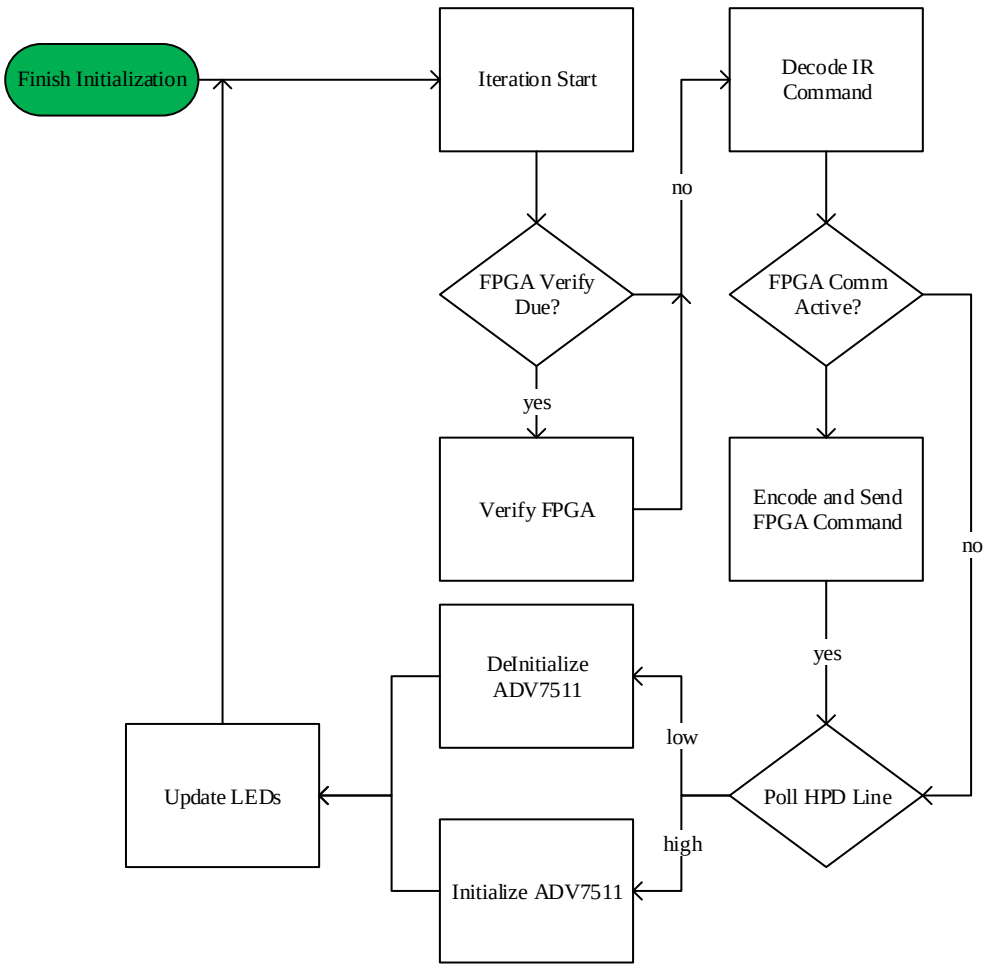


Figure 66: Microcontroller control processes summary.

6. Construction

6.1 Hardware

6.1.1 Acquisition

Digikey.com offers a large selection of discrete circuit components and discounts when parts are purchased in bulk. This supplier will be used for purchasing resistors, capacitor, integrated circuits, jacks and other discrete circuit components that will be installed on the PCB of the Super-Doubler. The PCB will be manufactured by OSH Park. This company manufactures two layer boards at $5.00 per square inch and four layer boards at $10.00 per square inch and includes three copies of the project.

A general enclosure will be needed to house the final PCB. This enclosure will be purchased at polycase.com. Modifications can be made to general cases to provide more structural integrity, improved heat ventilation and better line of sight for user interfaces.

A 12 volt ac-dc linear regulated wall adapter will be acquired from Jameco.com to provide power to the Super-Doubler. A linear adapter will be used to minimize voltage ripple on the main power rail of the PSD.

6.2 PCB Fabrication and Population

Surface mount circuit components will be temporarily applied to the PCB using solder paste and the assembly will be passed through a reflow oven for permanent installation. Through-hole components will be soldered by hand. Assembly and testing of the project will take place in the facilities provided at UCF such as the Senior Design and TI innovation labs where the appropriate equipment can be accessed. Since the circuit components used in this project are extremely small, problems may occur while soldering them to the PCB. If these problems become unavoidable, Small Batch assembly or Screaming Circuits will be contracted to assemble the PCB.

6.3 Software Tools

6.3.1 IDEs and Development Tools

For FPGA development, a primary motivation for choosing the 7 series Artix FPGA was support for Xilinx's Vivado Design Suite. Vivado is the evolution of the traditional Xilinx ISE development environment allowing more developer control. Vivado is a requirement for supporting the advanced features found on newer FPGA development boards and supports design using existing IP blocks supplied or purchased from Xilinx. It can also provide FPGA binary configuration files in a variety of formats as required by the microcontroller FPGA programming implementation.

For microcontroller development, several software tools will be used. STM32Cube provides a complete configuration utility for STM32 microcontrollers to generate configuration and startup files required for initialization of the MCU hardware. Example software and configurations are also provided to quicken the development cycle. For development, Keil MDK ARM will be used. Keil is one of the more powerful and streamlined IDEs for ARM development, and the STM32F0 microcontroller allows access to a license granting unlimited program code size when targeting an STM32F0 device. Programming will be done entirely in C for the microcontroller to avoid lengthy debugging and development processes in assembly. Additionally, all libraries selected are implemented in C.

6.3.2 Programmers and Flashing Utilities

The Keil MDK ARM development tool has complete debugging support for devices utilizing the STLINKv2 debugging interface. This interface and support will be used for real time programming and development of microcontroller code. For microcontroller flashing without use of the ARM SWD interface, ST provides a flash utility capable of

programming the microcontroller via its USART1 bus called ST Flash Loader Demonstrator. This program will be used with the system's USB programming interface to provide MCU code updates over that interface.

6.3.3 Software Libraries and Sources

Several software libraries and example packages are identified for use to reduce development time spent re-implementing solutions to common problems and in an effort to use more widely tested code. These libraries are primarily vendor supplied and aim to speed up development with the vendors' components. The libraries identified for use in this project as well as a brief description of their contents and purpose is found in Table 43.

Table 43: Libraries identified for use in microcontroller application development.

Library	Description
STM32F0 Standard Peripheral Library	Provides basic implementation of access to all peripheral devices in the STM32F0 line of microcontrollers including I^2C, SPI, USART, Timers, Interrupts, and other chip features
STM32F0 Eval Package	Builds on the STM32F0 Standard Peripheral Library to provide expanded features and example applications including IR interfaces and SD card device control
FatFs	Filesystem implementation for interaction with FAT formatted devices targeted at embedded devices. Requires some user device control implementation for functionality.

7. Prototype Testing

7.1 Microcontroller Interface Testing

As the microcontroller interfaces with every device in the system and is responsible for their initialization and control, it is important to test all of the interfaces and functions for correct results independently of the entire device assembly. For this purpose, a NucleoF070RB development board will be used in development, testing, and debugging of most application code on the microcontroller. The general process for testing interfacing will be similar for all devices and will be done with special purpose small footprint breakout boards for each IC to be tested. LEDs will be connected using a breadboard to the appropriate output LED pins used in full device operation to provide feedback for code functionality. In this manner, the functionality of the microcontroller in both initialization and control phases of the program can essentially be subjected to unit tests using specifically developed programs intended only for testing. This section details these test procedures.

7.1.1 Analog Devices IC Configuration

The three ICs from Analog Devices have similar enough interfaces that the testing of each can follow the same general structure. A test program will be written for the microcontroller for each IC. This test program will configure all writeable registers in the device to specific contents. The program will then read each register and verify that the read value matches the value known to have been intended for configuration. If all values are verified, the test will pass and all LEDs will be lit solidly. If the test fails on any register contents, all LEDs will steadily blink to indicate this failure. Note that for the ADV7511 this test procedure will only be used to verify configuration of those registers that are not reset when HPD is driven low.

7.1.2 ADV7511 HPD Dependent Control

A separate test program will be written for the ADV7511 HPD logic. Using the Nucleo and the ADV7511 breakout for testing the device will be initialized such that HPD is low (no device is connected). A known good device will then be connected to drive HPD high, and the OnHPDADV7511() configuration routine run when this is detected. The MCU will then read the registers similar to the tests for the rest of the Analog Devices suite and compare to the known programmed values. If any value fails, the test will fail and the MCU will blink the LEDs steadily. If all values are as expected, the MCU will drive all LEDs high and indicate success.

7.1.3 SD Interface

A test program will be written that is responsible for initializing the SD interface and filesystem, reading a specified file containing a specified binary string, and comparing the read in binary string to the expected value. On failure and success LEDs will be driven according to the same methodology used in the Analog Devices unit tests (blinking for failure, solid for success).

7.1.4 FPGA Programming and Interface

The FPGA interface used in the application is designed to provide continuous testing and verification throughout application execution. For discrete testing of this interface, a separate test program will be written that programs the FPGA using an FPGA configuration binary containing only the FPGA MCU interface implemented on the FPGA. This program tests both the MCU FPGA programming capability in loading a supplied FPGA binary and the communications specification between the FPGA and MCU after programming. This test has various points of failure necessitating more than two results. The various results are shown in Table 44 for convenience.

Table 44: FPGA interface unit test results.

Result	LED Feedback
Config File Not Found	FPGA Comm LED blinking, HPD LED solid
FPGA Programming Failure	HPD LED blinking, FPGA Comm LED solid
FPGA Comm Failure	Both LEDs blinking
Test Success	Both LEDs solid

These LED feedback configurations are also to be used in the full microcontroller application to indicate FPGA startup failure conditions.

7.1.5 IR Decoding

The IR interface will also be tested using a unique unit test program. Interfacing only with the IR LED, the supported commands will be sent via the IR remote control and read by the test program that continuously simply decodes incoming IR commands and then outputs an LED code for several seconds to indicate the command received. As there are only four supported IR remote commands, the test results will use a similar set of results to that in the FPGA interface test to indicate the command received. These results are summarized in Table 45.

Table 45: IR decoding test results.

Command Received	LED Feedback
Menu Up	FPGA Comm LED blinking, HPD LED solid
Menu Left	HPD LED blinking, FPGA Comm LED solid
Menu Right	Both LEDs blinking
Menu Down	Both LEDs solid

7.2 Target Companion Devices

When in digital video output mode, the Super Doubler can to be used with any TV which supports 480p using an HDMI connection. The VGA output can be used to connect to both CRT and LCD TVs. The VGA output provides a versatile RGB video source which can be easily transcoded into a format supported by the user's television set. The TV must support at minimum 480p in order to work with the Super Doubler when in VGA output mode.

The Super Doubler supports a wide range of video game consoles from the early Nintendo Entertainment System (NES), to the Sony Playstation 2. While all of the consoles listed in Table 46 support the three input video formats, the Super Doubler is optimized for the uses described in Table 46 below.

Table 46: Devices and supported signals.

	SCART-RGBS	VGA-RGB-HV	YPrPb
Nintendo Entertainment System (NES)	Yes	No	Yes
Super NES (SNES)	Yes	No	Yes
Nintendo 64	Yes	No	No
Sega Genesis	Yes	No	No
Sega Dreamcast	Yes	Yes	No
Sony Playstation	Yes	No	No
Sony Playstation 2	Yes	No	Yes

7.3 Requirements Specification Verification

A custom SNES game cartridge will be used to run test software on the SNES which will allow us verify that the device meets performance specifications. The test software allows for verification of color accuracy, frame processing time, and screen positioning. The display info function of our digital TV will be used to verify that the Super Doubler is outputting the correct resolution.

8. Administrative

This section documents project details that do not fit into the typical definition, research, design, prototyping, or testing phases of engineering project development. In many ways, these matters encompass the project's day-to-day development and housekeeping activities. Initial project budget estimates, initial project milestones and deadlines, and group organization, meeting, and communication structure are all found in this section.

8.1 Budget

The initial budget estimate for the project makes provisions for specific devices to be procured as well as estimates for devices typically obtained in bulk from various vendors and used throughout the project. These bulk items include discrete components such as resistors/capacitors and LEDs, as well as devices sold in similar quantities and used for generic purposes. Additionally, the budget takes into consideration enough components and fabricated boards to complete several prototypes in the interest of mitigating additional costs imposed by reacquisition in the event of board or component failure in the prototyping and testing stages. The budget estimate can be found below in Table 47.

Table 47: Initial project budget estimate.

Part Name	Unit Cost	Quantity	Total
Power System Components Set	~$30	3	$90
THS 7353 Video Buffer	$3	3	$9
AD1871 Audio Decoder	$10	3	$30
ADV7181 Video Decoder	$14	3	$42
ADV7511 HDMI Transmitter	$15	3	$45
Artix-7 FPGA IC	$40	3	$120
STM32F070RBT6 MCU IC	$2	3	$6
Miscellaneous ICs (Shift Reg, etc)	-	Varies	$50
Barrel Jacks (RCA, Power, etc)	$1	10+	$10
Miscellaneous Jacks (HDMI, VGA, SD, etc)	-	Varies	$50
Miscellaneous Components	-	Varies	$50
PCB Order Set	$40	1	$40
Enclosure	$10	1	$10
Remote/Receiver Components	$10	1	$10
SD Card	$10	1	$10
Development Boards and Tools	~$150	Varies	$150
Total			**$722**

8.2 Milestones

At the outset of the project a milestone chart was developed to provide estimates and deadlines for when various phases of the project should reach completion. These milestones apply to both the first and second semesters of the Senior Design curriculum. Most importantly, initial research and design are all completed in the first semester, and prototyping and testing are conducted throughout the second semester. The milestones adhering to this format are presented in Tables 48a and 48b.

Table 48a: Project milestones.

Task	Duration	Start Date	End Date
Research	**7 weeks**	**9/14/2015**	**10/31/2015**
- Video Protocols/Standards	2 weeks	9/14/2015	9/30/2015
- Communication Standards	2 weeks	9/14/2015	9/30/2015
- Audio Conversion	2 weeks	9/21/2015	10/7/2015
- FPGA	7 weeks	9/14/2015	10/31/2015
- - Selection	2 weeks	9/14/2015	9/28/2015
- - Logic and Interfacing	6 weeks	9/21/2015	10/31/2015
- MCU	6 weeks	9/21/2015	10/31/2015
- - Selection	2 weeks	9/21/2015	10/7/2015
- - Logic and Interfacing	4 weeks	10/1/2015	10/31/2015
- Analog/Video Filtering	3 weeks	9/21/2015	10/14/2015
- Analog-Digital Conversion	3 weeks	9/21/2015	10/14/2015
- Overlay Generation	2 weeks	10/7/2015	10/21/2015
- HDMI Transmitter	2 weeks	10/7/2015	10/21/2015
- Power Requirements	2 weeks	10/14/2015	10/31/2015
- PCB Considerations	2 weeks	10/14/2015	10/31/2015
- Enclosure (Late)	2 weeks	2/7/2016	2/21/2016
Design	**9 weeks**	**10/21/2015**	**12/21/2015**
- Filtering	4 weeks	10/21/2015	11/21/2015
- Analog-Digital Conversion (AV)	4 weeks	10/21/2015	11/21/2015
- Frame Manip. Logic (FPGA)	9 weeks	10/21/2015	12/21/2015
- Control Logic (FPGA)	9 weeks	10/21/2015	12/21/2015
- MCU Logic	9 weeks	10/21/2015	12/21/2015
- Overlay Generation	3 weeks	11/7/2015	11/28/2015
- Combined PCB	5 weeks	10/21/2015	11/28/2015
- Power System	3 weeks	10/28/2015	11/21/2015
- HDMI/Output Stage	4 weeks	10/28/2015	11/28/2015
- Enclosure (Late)	2 weeks	2/14/2016	2/28/2016

Table 48b: Project milestones.

Task	Duration	Start Date	End Date
Prototype	**5 weeks**	**2/7/2016**	**3/14/2016**
- ADC Breakout (Early)	3 weeks	9/21/2015	10/14/2015
- Output Stage Breakout (Early)	3 weeks	9/21/2015	10/14/2015
- Combined PCB	3 weeks	2/7/2016	2/28/2016
- Component Mounting	2 weeks	2/28/2016	3/14/2016
- Enclosure	1 weeks	3/7/2016	3/14/2016
Testing	**7 weeks**	**3/14/2016**	**5/4/2016**
- ADC Breakout (Early)	4 weeks	10/14/2015	11/14/2015
- Output Stage Breakout (Early)	4 weeks	10/14/2015	11/14/2015
- Finished Prototype	7 weeks	3/14/2016	5/4/2016
Documentation	**34 weeks**	**9/8/2015**	**5/4/2016**
- Initial Proposal	1 weeks	9/8/2015	9/15/2015
- SD1 Document	8 weeks	10/21/2015	12/15/2015
- Conference Paper	5 weeks	3/7/2016	4/14/2016

At completion of this paper, the group has completed the vast majority of research and design tasks expected to be necessary to begin prototyping and testing according to the design presented here. Some early proof of concept implementations have been tested, but only in the interest of exploring the viability of specific solutions or techniques. Between this time and formal prototyping, additional testing using breakout and breadboard setups is expected to help explore the creation of the first complete prototype and detect subtle problems not already known.

The next phases of the project, prototype construction and testing, are expected to take the bulk of the second part of the course. A first prototype should be completed during February to provide adequate time for redesigns and lead times necessary in reacquisition of parts. As mentioned in the budget estimate, enough materials will be ordered to construct several such prototypes. If significant problems are encountered in such crucial pieces of the project as the PCB design, these extra parts already on hand will help prevent any additional time lost from waiting on individual components.

Testing should begin at least one month prior to the final prototype demonstration and presentation date. This will allow as much time as possible in modification of reconfigurable elements of the project like FPGA and MCU logic to fully realize our requirements. As a key feature of our project is support for a wide range of devices and some of our testing criteria are subjective items like image quality, this time is also necessary to test many use cases and configurations. The devices necessary for this testing (various classic video game consoles) are either already on-hand or will be obtained prior to the beginning of prototyping.

8.3 Division of Labor

This project combines many aspects of electrical and computer engineering with their own levels of knowledge required for an effective design. As such, it is important to divide specific responsibilities and areas of expertise and research amongst the group members. Fortunately, this project contains subsystems interesting to particular group members such that labor division was quick and voluntary.

Stephen Williams has prior experience and research interest in FPGA development, and so readily took up the FPGA portion of project development. This includes the communications functionality needed to communicate with the microcontroller and various encoders, decoders, and transmitters as well as the video processing pipeline core to the project feature set. As he had the most prior knowledge of similar products, Stephen is also responsible for the video output stage including the HDMI transmitter and VGA output circuitry.

As the lone computer engineering major on the project, Kenneth Richardson has the most prior software development experience and so took up microcontroller programming and some peripheral device interfacing such as SD cards and USB connections. This development entails all internal microcontroller logic and implementation of any interfacing software required for communication between the microcontroller and support ICs.

With an interest in power and general electronics integration, John Shepherd heads up the power supply design for the fully integrated device and several peripheral devices. The inclusion of so many distinct ICs, a microcontroller, and more importantly an FPGA leads to an interesting power systems problem for the project. Additionally, John is responsible for user feedback and control peripherals including the remote control technology and output LED network.

Gilson Rodrigues took interest in the signal input stage of the project early on in project planning and so became responsible for those subsystems. Specifically, this is the input video signal filtering, input video signal analog-digital conversion and audio signal conversion and pass-through circuitry. This includes selection of the specialized ICs for each task and integration of the input stage into the final device in a user-friendly way.

As this project at its core requires integration and interfacing of a large number of components, member collaboration has proved extremely important in design of each subsystem. In an effort to keep duties clearly defined and as focused as possible, special attention is paid to design in pairs of any implementation details at an interface boundary. Group members responsible for specific subsystems work together to design any interface details between their subsystems. The most involved example is communication between Kenneth and Stephen about protocol between the microcontroller and FPGA.

Any design components that fall outside of any particular member’s area of expertise are worked out by the group as a whole. Examples of this include enclosure design and

sourcing as well as user experience considerations like the options presented for runtime user configuration.

8.4 Decision Making

Decision making is handled by the group as an entirety. As explained previously, every effort is made to keep subsystem boundaries as clear as possible. This reduces the number of disagreements or miscommunications between group members concerning internal implementation details and provides for a more efficient design and prototyping process with more time spent on implementation and less time spent deliberating over small details.

At subsystem interface boundaries, decisions are reached by consensus between the members involved in development of the interfaced subsystems. Consensus must be reached between these members, as a unilateral decision on implementation details, especially if the members stubbornly choose their own solutions, could be problematic and require significant redesigns later. If consensus is difficult to reach, group member's not directly concerned with the matter at hand will be consulted as well as outside advisors.

Any disagreements over design or implementation details in matters outside the scope of either individual subsystems or subsystem interface boundaries are made by the group as a whole. These matters could include budgetary problems, overall project design philosophies, housekeeping problems such as organization or formatting problems, or design of project components outside any one member's responsibilities.

All specific submissions over the course of the project's life must be agreed to be satisfactory by all project members before submission. This includes any deliverables like written assignments, papers, or prototypes as well as any materials used at important project milestones like the Critical Design Review or Final Presentation. This ensures participation and responsibility for all project matters by all group members.

8.5 Meeting Structure

Meeting structure will evolve over the project's lifecycle as project needs change. During the research phase of the project meetings are fairly sporadic and typically occupy time either before or after the lecture session as group member availability allows. At minimum this meeting time occurs once a week with all members present for at least one hour. This meeting time is used to ensure all group members are on the same page about where the project currently stands and what their current responsibilities and focus should be.

During the design phase of the project meetings necessarily become more frequent. These meetings are held as needed and typically outside of the normal minimum weekly meeting times between members concerned with the design material under consideration. This keeps with the same theme present in group organization and decision making that those members responsible for a subsystem are together the members responsible for any design

material at the boundary of their assigned subsystems. Weekly meetings are still conducted and will often be necessary to be several times per week with most or all members present.

During the prototyping and testing phase of the project group members will be in near constant communication. This phase of the project will require by far the most collaboration among members and the largest time commitment. Any delays or miscommunications could cause a week or more lost in lead times or other scheduling problems outside of the group's direct control. Multiple weekly full group meetings will be held to ensure proper pace of progress and every effort will be made for all group members to be available for work at consistent times. During this time meetings will resemble more of scheduled work times than the meetings in previous phases.

8.6 Communication and Remote Collaboration

As with any collaborative engineering effort, design or otherwise, communication between members is vital to our project. Every phase of the project including research into technologies and components, selection of components and strategies for solving the project's technical challenges, and especially design of the various subsystem interfaces between members' focus areas requires extensive collaboration between responsible members. Physical meetings are not always possible nor the best means of working together. Additionally, half of the group lives quite far from the campus area. Therefore many tools must be used together towards group collaboration so work can be accomplished remotely.

Group messaging is the first line of communication for our group. Using group messaging services such as GroupMe, Google Hangouts, and email or private text messaging we coordinate our schedules, meeting times, and immediate needs. By using group services, group members otherwise not necessary to be included in a particular meeting are still made aware of the rest of the group's activities. This has the benefit of cataloguing group activity over time and providing periodic status updates, essentially answering some questions between group members before they even need to be asked.

Of course, housekeeping-type communication such as scheduling meetings is used mostly to facilitate real collaboration in preparing the project's various deliverables. Over the project's life, an enormous amount of important information is found and needed to be shared. Also, in the design and prototyping efforts much brainstorming and many iterations of project documentation are produced and need to be worked on and shared between members. To this end, we make extensive use of Google Drive and the various office applications it offers. Early drafts of many pieces of documentation are made and later finalized using Microsoft Office tools (thanks to their superior formatting tools). Web apps like Draw.io also provide tight integration with Google Drive and make it easy to quickly make up diagrams that are immediately available to the whole group before either use in a deliverable document or in more sophisticated drawing programs.

Collaboration on more technical files like source code for the microcontroller implementation, Verilog or schematic capture files for the FPGA, and various schematic

and PCB files is more efficiently accomplished via a proper source control platform. To this end we plan to use Git and one of the many online Git hosts like Github. These kinds of resources typically undergo many changes during a project's lifecycle. By taking advantage of a version control tool like Git, we can preserve the entire series of changes applied to these files. Remote hosts like Github also supply built in documentation, communication, and management features like wiki pages, markup/down supported Readmes, and issue tracking. This makes collaboration much more organized than the classic solution of passing around continuously changing source files by shared drives or email.

Due to the scattered location of our group members and scheduling conflicts, this remote group collaboration has been extremely important. Thanks to the many web based tools available today, much of the project outside of physical construction and testing can be accomplished without in-person group meetings. This has been the predominant method of communication and collaboration for our group, and in our experience in many ways been superior to long face-to-face meetings.

Appendix A. Copyright Permissions

|==========
| Staff Comment 2015-12-09 08:41:39 PST
| By: Cristina C
|
| Dear Gilson,
|
| Thank you for your inquiry.
|
| Please feel free to use the picture referenced in the mentioned
| link. Please reference Maxim website as the source on your
| report.
|
| Please don't hesitate to contact us should you require further
| assistance.
|
| Best regards,
|
| Cristina
| Customer Operations
|
|==========
| Submit Request 2015-12-09 08:38:18 PST
| By: rodrigues.gilson@knights.ucf.edu
|
| Good Morning,
|
| I am Gilson Rodrigues EE student at University of Central
| Florida. I would like to know if it is ok to use a picture from
| your website? I will be using it as part of my Senior Design
| project report.
|
| Thanks
| Gilson Rodrigues
|
| In the below link Figure 2
| https://www.maximintegrated.com/en/app-notes/index.mvp/id/1184
|
|==========
| REFERENCES
|
| Referring Url:
| https://www.maximintegrated.com/en/app-notes/index.mvp/id/1184
|
| Maxim Support Center:
| https://support.maximintegrated.com/
|
| Maxim Home Page:
| http://www.maximintegrated.com/
|
|==========

Fwd: Picture permission

Gilson Rodrigues
To: shepherdjt@hotmail.com;

Reply
Wed 12/9/2015 7:51 AM

---------- Forwarded message ----------
From: **RetroRGB** <retrorgb@gmail.com>
Date: Tue, Dec 8, 2015 at 1:45 PM
Subject: Re: Picture permission
To: Gilson Rodrigues <rodrigues.gilson@gmail.com>
Yes, absolutely. To make it "official": You have my permission to use the file: http://www.retrorgb.com/images/240p-4k.png

Take care.

- Robert Neal

On Tue, Dec 8, 2015 at 1:37 PM, Gilson Rodrigues <rodrigues.gilson@gmail.com> wrote:

> Good afternoon,
>
> I am Electrical Engineering Student at University of Central Florida and I am interested in using one of your pictures displayed in your website www.retrorgb.com.
> The picture in on the tab "240p" and the picture is under "Difference of each resolution"
> I would like to know if that it is ok with you
>
> Thanks
> Gilson Rodrigues

To: <cic.americas@analog.com>
cc:
From: Gilson Rodrigues <rodrigues.gilson@gmail.com>
Date: 12/07/2015 09:05:17 PM
Subject: picture permission of use

Good evening,

I am an Electrical Engineering student at University of Central florida and I am interested in using some pictures from your website. Some of the pictures I would like to use is from the parts, which are pictures showing diagrams, pin layouts and table:

* AD1871
* ADV7181

Thank you!
Gilson Rodrigues

Gilson,

You hereby have permission to use product images from the ADI website in your coursework projects, as long as the ADI source is cited.

Thanks, and good luck with your project.

Regards,

Jim Surber

Technical Content Manager

Analog Devices, Inc.

Gilson Rodrigues

To: shepherdjt@hotmail.com;

Reply

Wed 12/9/2015 10:35 AM

---------- Forwarded message ----------
From: **Tim Green** <tim.green@tikilive.com>
Date: Wed, Dec 9, 2015 at 11:31 AM
Subject: RE: picture permission of use
To: Gilson Rodrigues <rodrigues.gilson@gmail.com>

Hello and yes that is fine. 🙂

Kindest Regards,

Tim Green

Founder

TikiLIVE.com

305-289-4557

TikiLIVE Movie – what we do!

From: info@tikilive.com [mailto:info@tikilive.com] **On Behalf Of** Gilson Rodrigues
Sent: Wednesday, December 9, 2015 10:49 AM
To: info@eyepartner.com
Subject: picture permission of use

Good morning,

I am Electrical Engineering Student at University of Central Florida and I am interested in using one of your pictures displayed in your website http://www.thehdstandard.com/
The picture of the RCA connection.

Let me know if this ok.

Thanks

Gilson Rodrigues
http://www.thehdstandard.com/streaming-technology/audio-connections/

---------- Forwarded message ----------
From: **Gilson Rodrigues** <rodrigues.gilson@gmail.com>
Date: Wed, Dec 9, 2015 at 10:28 AM
Subject: Picture permission
To: mhi@penguin.cz

Good morning,

I am Electrical Engineering Student at University of Central Florida and I am interested in using one of your pictures displayed in your website http://martin.hinner.info/vga/scart.html
The picture of the SCART 21 pin configuration "Solder Side"
Let me know if this ok.

Thanks
Gilson Rodrigues

Permission to use figures

JS John Shepherd

To: copyrightcounsel@list.ti.com;

Reply all

Mon 12/7/2015 11:20 AM

Sent Items

Dear Texas Instruments Copyright Counsel,

My name is John Shepherd. I am a student at the University of Central Florida in the Electrical Engineering and Computer Science program. I am currently taking the Senior Design course and am writing the project document. In this document,
my group members and I walk through the design methods for the systems used in the project. The final document will be printed and bound and delivered to the professor for review. The document will also be displayed on an original website that the team must create in the future.

I am requesting permission to use information documented in data sheets for Texas Instruments parts, specifically reference design figures.

The reference designs for the following parts:

TPS563200
TPS7A8101
TPS40305

The purpose for the reference design figures is to demonstrate the use of the part and how it will be implemented in the final design. It is also intended to help the reader understand the modifications that will be made.

Sincerely,

John Shepherd.
B.S.E.E undergraduate.
University of Central Florida.

SU support@ti.com

To: John Shepherd;

Reply all

Wed 12/9/2015 10:33 AM

Hello John

Thanks for contacting TI Customer Support.

The permission to use some of our figures is covered in our terms of use, please see the direct link below for further assistance

http://www.ti.com/corp/docs/legal/termsofuse.shtml#access

If you have any questions or concerns, please feel free to contact us.

Regards,
Chinenye Ndubuokwu
TI Customer Support
Americas Customer Support Center
512-434-1560

Fw: Reference Circuit for ADV7511 FIGURE 32

Stephen Williams

To: shepherdjt@hotmail.com;

Reply

Wed 12/9/2015 12:13 PM

Inbox

From: Stephen Williams
Sent: Thursday, December 3, 2015 9:22 AM
To: ATV_VideoTX_apps@analog.com
Subject: Reference Circuit for ADV7511

Hi AD,

I'm a senior undergraduate student at the University of Central Florida requesting information for my senior design project. I would like to request a complete set of reference schematics and PCB layout for the ADV7511 as mentioned in page 54/58 in the hardware users guide.

Thank you,
Stephen Williams

Figure 4

Stephen Williams

To: shepherdjt@hotmail.com;

Inbox

Photos

User details:

From User: specialty-av (27428)

99.9% Positive Feedback

Member since Jan-24-00 in United States

Location: FL, United States

Activity with specialty-av (last 90 days): I have bid on **0** items from specialty-av

Activity with specialty-av (last 90 days): specialty-av has bid on **0** of my items

Dear steph-willi,

Hello steph-willi,

Permission granted.

Thank you for your courtesy.

Arthur

specialty-av

- specialty-av

From: steph-willi
To: specialty-av
Subject: steph-willi has sent a message
Sent Date: Dec-07-15 08:18:31 PST

Dear specialty-av,

I'm an electrical engineering student at the University of Central Florida. I would like to request permission to use a picture of your product found here http://www.ebay.com/itm/NEW-SCART-RGB-to-YUV-Component-Video-Converter-Scaler-/221156873851?hash=item337dfa167b:m:mL77q83PIBdVHyF2RvMlcww in my senior design document. I'm using this image to reference existing devices.

My group is building a video scaler. We will be needing to purchase one of your units in spring for testing the input.

Thank you,

- steph-willi

Stephen Williams
To: shepherdjt@hotmail.com;

Reply
Wed 12/9/2015 12:17 PM

Inbox

From: HDMI Admin <admin@hdmi.org>
Sent: Friday, December 4, 2015 8:14 AM
To: Stephen Williams
Subject: Email sent from HDMI website contact form - General Question (10215)

Thank you for contacting HDMI Licensing, LLC. We are reviewing your submission and will respond when we have more information. Your comments and feedback are welcome and appreciated.

First Name:	Stephen
Last Name:	Williams
E-mail Address:	stephen.williams@knights.ucf.edu
Affiliation:	Other
Company Name:	UCF
Country:	United States
Industry:	Other
Subject:	General Question
Message:	Hi, I'm a senior undergraduate at the University of Central Florida working on my senior design project. I would like to request permission to use the Figures in the HDMIspecification1.3 documentation found here http://www.microprocessor.org/HDMISpecification13a.pdf I would like to use the figures in my Senior Design project report. Thank you, Stephen

Gilson Rodrigues <rodrigues.gilson@gmail.com> 10:35 AM (3 hours ago)

to sales

Good morning,

I am Electrical Engineering Student at University of Central Florida and I am interested in using one of your pictures displayed in your website http://www.bluejeanscable.com/
The picture of the Component video.
Let me know if this ok.

Thanks

picture in the link below
http://www.bluejeanscable.com/articles/componentvideocable.htm

---

Blue Jeans Cable <sales@bluejeanscable.com> 1:58 PM (6 minutes ago)

to me

Yes, no problem -- you're welcome to use it (with attribution).

Thanks,

Kurt
BJC

From: Gilson Rodrigues [mailto:rodrigues.gilson@gmail.com]
Sent: Wednesday, December 09, 2015 7:36 AM
To: sales@bluejeanscable.com
Subject: Picture permission Component Video

picture permission of use

Gilson Rodrigues <rodrigues.gilson@gmail.com> 10:54 AM (5 hours ago)

to reserved

Good morning,

I am Electrical Engineering Student at University of Central Florida and I am interested in using one of your pictures displayed in your website http://www.cablemagic.com.au/
The picture of the SCART to RGB connector
Let me know if this ok.

Thanks
Gilson Rodrigues

http://www.cablemagic.com.au/scart-to-component-cable-1-5m-1.html

RE: picture permission of use(Case#: C15LG0052) Inbox x

Surber, Jim <Jim.Surber@analog.com> Dec 8 (1 day ago)
to me, CIC

Gilson,

You hereby have permission to use product images from the ADI website in your coursework projects, as long as the ADI source is cited.

Thanks, and good luck with your project.

Regards,

Jim Surber
Technical Content Manager
Analog Devices, Inc.

To: <cic.americas@analog.com>
cc:
From: Gilson Rodrigues <rodrigues.gilson@gmail.com>
Date: 12/07/2015 09:05:17 PM
Subject: picture permission of use

Good evening,

I am an Electrical Engineering student at University of Central florida and I am interested in using some pictures from your website. Some of the pictures I would like to use is from the parts, which are pictures showing diagrams, pin layouts and table:

* AD1871
* ADV7181

Thank you!
Gilson Rodrigues

TI product parts picture permission

Gilson Rodrigues <rodrigues.gilson@gmail.com> 9:20 AM (7 hours ago)
to copyrightcouns.

Good morning,

I am Gilson Rodrigues EE student at University of Central Florida. I would like to know if it is ok to use pictures of the ICs/parts from your website products? I will be using these pictures in my Senior Design report.

Thanks

...

Gilson Rodrigues