Flashback

Senior Design II

Group 22:

Marlon De La Cruz

Rene Martinez

Trenton Reed

Marlon Smith

Table of Contents

Table of Figures

Table of Tables

1.0 Executive Summary

Flashback (patent pending) is a solution to the television viewer's problem of missing the action in their favorite programs. Flashback is a feature designed to detect the end of a commercial break in real-time. The user is able to enable our feature on a primary television signal of their choosing. After enabling Flashback, the user may freely roam to other secondary television signals without the worry of missing any action on the primary program. While the feature is enabled, Flashback will analyze the incoming primary television signal to detect the end of the commercial break on the primary program. When our feature determines the commercial break is ending, the user is prompted and asked if they would like to remain on the secondary program or if they would like to resume the primary program. If the user wishes to remain on the secondary program, then Flashback sets a marker on the recorded data for the primary program, so the user may return to the moment in time when the commercial break ended. If the user does return to the primary program at the end of the commercial break, Flashback records the primary program, and then the user is able to resume action on television the program from end of commercial. In addition to the recording feature, Flashback remains active until the user specifies another television signal or disables Flashback on the original primary program.

In order to accomplish Flashback, we developed a device that is a simple additional connection to the user's existing cable box, an over-the-air television programming antenna, or a satellite television receiver. Our device is able to record two television programs. The first being the channel of which Flashback is enabled and the second being the channel of which the user is watching while the primary program is recorded and analyzed. The secondary signal could be a single channel or it could be the user flipping through a number of channels. To analyze the television signal and detect commercials we used a Gumstix Overo® TidalSTORM computer-on-module and a cape board loaded with high-level software. To control the Flashback device, our group uses an infrared remote control that contains a user friendly 7-segment display to confirm user commands.

Our goal in the design of Flashback was to enhance the viewing experience while creating a device that is portable, low-powered, easy-to-use, real-time, and most important of all, accurate. We also plan to incorporate the elimination of commercials within the recording of these programs. This feature will provide users with uninterrupted, recorded playback removing the need for users to fast forward through commercials. This aspect of recording will also save space for recorded programs due to the elimination of the commercial blocks that every television program contains. Future revisions of the project will target homes, families, or sports bars when multiple games are being played in an allotted time.

2.0 Project Description

This section will outline the motivation and goals of the intended project, while describing the general functionality of the device we plan to build and execute. It will also discuss the objectives desired through the pursuit of Flashback.

2.1 Project Motivation and Goals

Flashback is a device that increases the viewing pleasure of anybody sharing a single television with different interests in television programming. Although there are already devices that detect commercials and allow for instant transitions between channels, they aren't as accurate as Flashback nor do they function with live stream video. Flashback's main goal is to detect the transition between live television and commercials so primary or secondary channels can be switched instantaneously while maintaining a high accuracy to allow viewers to pick up where they left off before the commercial break.

The idea for this device was conceived through the viewing of a sporting event on television while other individuals were present that were not interested in watching the sporting event. Since sporting events have erratic commercial breaks, the manual switching from the channel hosting the sporting event to any other channel can potentially lead to the missing of a big play or a certain amount of the time of the game, which in turn, leads to frustration. This frustration can fall true for any genre of television. Digital video recorders actually record every channel a viewer watches on its main memory; this action allows users to rewind and fast forward recorded live television, however, the catch is that once they switch channels, that recording is lost unless they set that channel to record. Using this current technology, there would be no way for a user to watch a part of the program that they missed unless that channel is set to record. With this device, we remove the chance that users could miss a part of their program since the device knows when to switch back to the channel they set as their primary.

With the new rise of smart technology, almost everything is becoming smart nowadays such as smart cars, security, and phones; why not introduce smart to your DVR? Flashback embodies the idea of 'smart' through self-detection of commercial breaks, and since the device can detect these commercials why not eliminate them from recordings? Although DVR's record programs effectively, the recording algorithm of a DVR does not discriminate between actual scheduled programming and commercials. Without this distinction, users have to fast forward through commercials that could have been eliminated. In many cases, users will forget that they are watching a recording and will spend time watching commercials.

We created a device that will allow the user to take back their living room from advertisements that interrupt a user's free time to enjoy entertainment.

2.2 Objectives

Flashback's design utilizes many low power integrated circuits to provide the lowest possible power supply without sacrificing device functionality. Our device uses American standard wall outlet AC power translated to DC power using an external converter. Input DC voltage for safe operation on Flashback device is 5V running at 2.5 A. The device runs on an embedded operating system with a CPU dedicated to storing the current channel being watched on the main memory while running the appropriate algorithms for determining if the channel has just reached a commercial break. All the processes necessary in achieving these tasks to be threaded especially the frame by frame operations. Multi-threading is also necessary for detecting user commands that require other tasks to be run during commercial detection which will remove any lagging associated with process switching.

Flashback implements an algorithm that is at least 95 percent accurate in detecting commercials on both channels. When the algorithm detects a commercial from the on-screen display, the GUI Overlay of the software notifies the user that a channel's commercial break has ended and will either automatically change channels or require users to give the device the "okay" to switch. The algorithms are streamlined enough to prevent any lag or processor lock so that users don't experience any inconvenience from using the device.

The on-screen display of the device is similar to what users are familiar with, while limiting the input needed from the user. The display is adaptable for different television sizes. When the user inputs the commands they want executed, the options easy to understand and the different classes of options are under the correct menu so users can find them. The main goal for on-screen display was to provide effortless use of Flashback because the time spent watching television should be enjoyable; users won't want to use a device that's too complicated.

The device is able to record a user's preferred channel while maintaining responsiveness to the user. This functionality of not recording commercials provides users with a richer viewing experience since it eliminates the need to fast forward through commercials. The recording portion of the algorithm receives a signal from the commercial detection portion of the software as to whether a commercial break has begun or ended. The recording portion pauses when a commercial break is detected and resumes when a commercial break has ended.

Overall, the device's main objective was to provide the user with the best possible viewing experience. Commercials and advertisements are a hassle for users to watch. Our group provided the user with a way to avoid these advertisements during live and recorded programming.

2.3 Project Requirements and Specifications

This section will discuss the project requirements, along with the hardware and software specifications of Flashback. It will also numerically define “real-time” and “live” programming for Flashback

2.3.1 Project Requirements

Flashback is a device that’s similar to Digital Video Recorder with some extra functions to allow users to skip commercials. The Flashback device itself is low powered, responsive to user input, efficient, and easy to use. The device needs dual-television tuners to allow the software access to two channels. Without the dual tuners, the algorithm that pivots between channels wouldn’t work. The processor in the device is powerful enough to handle parallel operations on matrices with at least 64 by 48 pixels in a quick manner. Each of these pixels will contain an RGB color code of 24 bits that will range from 0 to 16777216. The processor must be able to operate on a 64 by 48 frame, which includes pixel values that fall within this large range, and are able to maintain response in real-time.

Flashback contains an on-screen display that provides users with a familiar layout that contains functionality similarly to what one would see using a DVR. This interface may also request user input from a remote control, mouse, or keyboard. The interface requires minimal user input to function properly.

Every layout for the on-screen display contains font that’s legible from the average distance that user’s watch television from, which is about 10 to 15 feet away. The on-screen display efficiently streams frames from compressed MPEG-2 files. The MPEG-2 file format is a compression format, widely used by television programming providers to supply their viewers with high definition digital television. Without support for MPEG-2 files, Flashback would not be able to provide its user with expected or sufficient television programming.

Flashback supports the recording of channels and more importantly the recording of programs without commercials. The algorithm uses codecs to compress and encode these streaming videos into MPEG-2 files for storage onto the device’s drive for playback. To support the removal of commercials from recordings, the processor is able to multi-thread, so the transition from the advertisement detection algorithm to the recording algorithm is quick and efficiently accomplished. Without the use of multi-threading, Flashback’s main feature of quickly detecting and deleting commercials from live television would be extremely slow. Recording a user’s program without commercials provides the user with an uninterrupted program, thus, enhancing their viewing experience.

2.3.2 Minimal Technical Hardware Specifications

Below our group has outlined our initial hardware requirements and expectations for Flashback

Processor

- Must be an ARM type
- Use ARMv6 Architecture or Higher
- DDR 2 Support
- Built in cache
  - L1 is a must
  - L2 is optional
  - Size of either will be based on based on processor
- Onboard Digital Signal Processor
- Onboard Graphics Processor
  - Support at least two video capture channels
  - Support output of High Definition Resolution
  - Support scaling MPEG-2 Videos
- If it is Multi-Core
  - Maximum 4 cores
  - Each core must have a clock of at least 400 MHz
- Single Core
  - Clock rate of at least 1.0 GHz
- Low Power
  - Operating voltages of either 1.8 V or 3.3 V
  - I/Os, with exception of USB and DDR, operate on 1.8 V or 3.3 V
- At least two external memory interfaces for RAM and Storage Device
- Floating Point for increased dynamic range and precision
- LCD Controller
- At least two master/slave I^{2}C buses
- SATA Controller
- At least 100 configurable pins for peripherals

RAM

- DDR 2
- 512 Mega Bytes
- Clock rate of 800 MHz

Storage

Hard Disk Drive

- At least 5400 RPM

- A Sizeable Cache
- Low power usage
- SATA 2 interface
- Internal/Laptop Form Factor
- 80 GB

Solid State Drive

- Low power usage
- SATA 2 interface
- Internal/Laptop Form Factor
- 32 GB
- Multi-Layer Cell

Connectivity

Infrared

- RC5 protocol

Bluetooth Protocol

- Version 2.1
- Version 3.0
- Version 4.0 (Low Energy)
- Must have open source stack

Near Field Communication (NFC)

- Must have open source stack
- Low profile antenna
- Support card emulation

WLAN

- Support for protocols:
  - Wireless N
  - Wireless B/G
- 2.4 GHz Band
- 5 GHz Band

Dual Television Tuners

- Support ATSC
- Support NTSC

Analog to Digital and Digital to Analog Converters

Internal Interfaces

- SPI
- UART

- SATA 2.0

External Interfaces

- JTAG for hardware debugging
- GPIO Pins for future upgrades
- USB
    - Prototype with Version 2.0
    - Production will hopefully use 3.0
    - Multiple hubs
- LCD
    - 7 Segment Display
    - Receiver LED
    - Channel Display upon input
    - Clock for time display
    - Potential Character display

External Video Interfaces

- High Definition Multimedia Interface (HDMI)
- Composite and Component Video

External Audio Interfaces

- Composite and Component Audio
- Optical Audio

3.0 Research Related to Project Definition

This section will discuss all relevant information, reference designs, costs, potential critical design options, and algorithms that may have any relation or contribution to the design and definition of Flashback

3.1 Digital Video Recorder

We will first discuss the digital video recorder, since it is the most similar device to the basis of Flashback, which we will expand and implement out intended design as a basis for or device. It appears to be an ideal reference design and algorithm basis.

3.1.1 Overview

A digital video recorder (DVR) records video provided by a digital television service provider onto a disk drive, USB, SD memory card or other mass storage devices. The incoming signal will come from an antenna, cable, or satellite device. Then it will pass through a tuner in which the user will be able to select the desired television program. Next, the signal will be sent to an MPEG-2 encoder to convert the stream to a format desired by DVR device. In the case of digital transmission from broadcast companies, no MPEG encoders will be necessary because satellite and cable companies will have converted the signal already. From the encoder the signal will be sent to two places: a storage device and a decoder, which will convert the stream and send to the television for user viewing. Flashback will need to alter the data flow slightly to enable our commercial detection algorithms, but our device will still be able to air television programs in a live feed (Strickland and Bickers).

A typical DVR device will usually operate on a highly modified Linux operating system. This operating system will reside on a local storage device, and the storage device will also have enough room to record television programs, provide a buffer for live broadcasts, and it might even have space to expand the program in the future. To record programs the device should account for approximately one gigabyte per hour of television in a basic setting and four gigabytes for a high-quality setting (Strickland and Bickers). Furthermore, DVRs require a buffer to handle the possible delays in receiving video stream information. The buffers are able to provide a consistent playback for the viewer and the user won’t notice any delays that do happen.

3.1.2 Features

DVRs have many different features that enhance the viewing experience in comparison to only being able to watch a live feed on a single program. One of the many features that DVRs typically have is the ability to record different programs on different channels at the same time. To enable the device to perform these tasks, the DVR will have multiple television tuners. Another simple but impressive feature about digital video recorders is their ability to pause live television. DVRs actually don’t “pause” the incoming signal until one is ready to view it. The device actually acts like a buffer and is constantly recording the television program. By pressing “pause,” the user actually freeze-frames the current image being displayed and marks the location in memory of when the signal was “paused.” When the user hits “play” the recorded program begins playing again. Furthermore, some DVR devices have the ability to record up to 300 total hours of television programs depending on the video quality (Strickland and Bickers).

3.1.3 Costs of DVR

DVRs usually have a monthly fee associated with their use. The reason for an ongoing payment system is because the devices will have to open up communications with a server at the broadcast company to download new program guides (Strickland and Bickers). The pricing of DVRs will depend heavily on the amount of applications desired. They could range from $50 to approximately a couple thousand dollars.

3.2 FPGA vs. DSP

This section will discuss the pros and cons of choosing between a FPGA and a DSP for digital signal processing hardware and all its components. The applications we want to include in Flashback will play a huge part in this design decision

3.2.1 Overview

There are two front runners for digital signal processing hardware, Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs).
Digital Signal Processors are essentially a specialized microcontroller whose characteristics are optimized for high-speed processing applications. DSPs take real-world signals that have been digitized and then mathematically manipulate them for specific applications. DSPs offer a combination of arithmetic operators, memory handling, instruction set, parallelism, and data addressing are the key difference between DSPs and other kind of processors. Digital signal processors offer many benefits including the ability to process data in real-time (Skolnick and Levine).

A Field Programmable Gate Array is a programmable semiconductor device that contain a matrix of Configurable Logic Blocks (CLBs) connected with programmable interconnects. These devices are not built for a particular design. They can, in fact, be programmed to a desired application or functional requirement. If the desired application or functional requirement changes, the FPGA may be reprogrammed to implement a new design. These designs can be done very late in the design cycle even after the device has been deployed to the field. FPGAs at the minimum have CLBs, interconnects, select inputs and outputs, memory, and clock management blocks ("FPGA vs. ASIC").

The purpose of Flashback is to process video stream information in real-time while interacting with the user. To accomplish this, our device must be able to complete analysis of a current sample before the next sample arrives (Skolnick and Levine). Therefore, DSP or FPGA selection will use performance as its number one priority with power consumption, and cost as secondary priorities. All of these aspects weigh heavily on our groups decision to utilize one device over the other.

3.2.2 Performance

Both Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs) are able to be reprogrammed frequently and at possibly crucial times in the design process. However, if the design specifications change an FPGA is able to adjust to the new modifications faster than the DSP. With a DSP, there may be time spent wasted on obtaining a new component if the design specifications change (Wain).

Furthermore, FPGAs are parallel in nature unlike processors. This allows FPGAs to execute commands without competing for the same resources. In order for FPGAs to truly work in parallel the hardware contains dedicated sections of the chip and can execute without any influence on the other logic blocks. They are able to control inputs and outputs at the hardware level to provide a faster response time (“Introduction to FPGA Technology: Top 5 Benefits”). FPGA’s flexible ability makes the hardware more desirable for prototyping situations.

FPGAs are typically better with tasks that involve fixed point data, and they are not that good with high precision floating-point arithmetic (Wain). DSPs are able to handle more complex algorithms and varying data sizes much better than FPGA. Since image processing focuses on “regions of interest” in an object, those regions could be varying sizes and therefore the processing will be more complex (“Choosing FPGA or DSP for Your Application”).

Before an FPGA is programmed, it knows nothing about how to communicate with other devices which allows increased flexibility but also increased complexity in programming the device. Since DSPs are geared more toward specific applications, the processor is able to recognize the peripherals and how they communicate with other devices. In addition, the software development can be done in a higher level programming language, such as C (Wain). This programming environment will make it easier to incorporate more complex algorithms for the nature of our group’s project. Both DSP and FPGA offer libraries for basic signal processing, such as Finite Impulse Response filters and Fast Fourier Transforms (“Choosing FPGA or DSP for Your Application”).

3.2.3 Power Consumption

FPGA’s highly flexible architecture comes with a disadvantage of more energy consumption. They have increased energy consumption because FPGAs have more gates, more silicon area, and more routing resources. DSPs are tailored for efficient implementation, and therefore, they can have lower chip-level energy consumption. Berkeley Design Technology, Inc. (BDTi) performed an analysis of power consumption on demanding DSP applications. Their results showed that FPGAs consume approximately 10 Watts, while high-end DSPs consume roughly 2-3 watts. These numbers work in favor of overall chip-level consumption but since FPGAs can support 10 to 100 times more channels the energy consumption per channel is significantly lower than that of DSPs. Our group looked at the chip-level energy consumption because Flashback’s algorithms will not need to utilize more channels than the many offered through FPGA (“FPGAs vs. DSPs: A Look at the Unanswered Questions”).

	Device Family	Device Cost (Dollars)	Max 32-bit MMAC	Cents / MMAC
DSP	C5000 Fixed Point DSPs C2000 Controllers	<10	25 - 300	1.8 - 46
	DaVinci Digital Media Processors C6000 Fixed Point DSPs C6000 Floating Point DSPs C5000 Fixed Point DSPs C2000 Controllers	10 - 30	50 - 1200	1.6 - 35
	DaVinci Digital Media Processors C6000 Fixed Point DSPs C6000 Floating Point DSPs C5000 Fixed Point DSPs	30 - 100	50 - 1440	3 - 44
	DaVinci Digital Media Processors C6000 Fixed Point DSPs C6000 Floating Point DSPs C5000 Fixed Point DSPs	100 - 300	266 - 2000	5.8 - 48
	C6000 Fixed Point DSPs	300 - 330	2000 - 2400	13.4 - 14.4
FPGA	Cyclone II Cyclone III	10 - 30	270 - 660	1.4 - 7
	Cyclone II Cyclone III	30 - 100	380 - 1900	2.8 - 20
	Cyclone II Cyclone III Stratix III	100 - 300	1000 - 4500	2.9 - 34
	Cyclone II Cyclone III Stratix II	300 - 1000	1000 - 8300	4.2 - 47
	Stratix II Stratix III	1000 - 3000	3000 - 11000	20 - 100
	Stratix II Stratix III	3000 - 10000	4000 - 11000	20 - 130

Table 3 – 1: Information provided from Afra and Kapadiya's article "Making design choices between DSP and FPGA"

3.3 Digital Signal Processor

Our group decided that we will use a digital signal processor over an FPGA based upon the requirements of Flashback, the groups skill set, and desire for experience with DSP hardware. Due to the real-time nature of Flashback, our DSP selection will use performance as its number one priority. In analyzing the performance of a DSP we will look at the arithmetic format, speed, memory, peripherals, and power.

3.3.1 Arithmetic Format and Data Width

A key characteristic in DSP selection is whether to use fixed-point or floating-point arithmetic. Fixed-point arithmetic essentially means that numbers are represented as integers or fractions within a fixed range. Floating-point arithmetic means that values are represented by a mantissa (decimal part of a logarithm) and an exponent, i.e. mantissa x 2 ^ exponent ("Choosing a DSP Processor"). To determine whether the systems need for either floating-point or fixed-point the designer must first look at the following characteristics: data width, dynamic range, ease of implementation, and cost.

Floating-point devices typically have a larger data width, precision, and ease of implementation than fixed-point. The data width will have an impact on the size of the chip, number of pins, and the size of external memory. With floating-point arithmetic the designer will be able to implement greater precision. When the DSP performs a calculation, that number may be rounded to the nearest value based upon the format. Due to the larger data width, the floating-point device will not need to round or truncate the value as much and this will prevent unwanted quantization noise derived from representing analog signals in a digital format.

Floating point devices are easier to program because the programmer will not have to worry about number scaling prior to arithmetic operations to avoid rounding and truncating errors. A wider dynamic range means that the circuit will be more complex and therefore will require a larger silicon chip. The wider dynamic range of floating-point DSPs will typically cause them to be more expensive and have a higher amount of power consumption ("Choosing a DSP Processor").

Fixed point devices have speed, efficiency, and cost on their side. Fixed-point devices, with a smaller data width, do not require as large of a chip or pin count and this will reduce the cost. However, these devices are more particular when they are programmed. The designer must be aware of data flow during all stages of processing since at some points they may need to carefully scale the signals to ensure numeric precision within the limited dynamic range. Typical design practice is to choose a device with the smallest word size that can still support the fundamental application. The clocks of fixed-point systems can typically support greater speeds than floating-point. In many high-volume, embedded applications a fixed-pint device is chosen for the lower cost and power consumption ("Choosing a DSP Processor"). Furthermore, a decoder in the format of MPEG-2, MPEG-4, or JPEG-2000 utilizes decoding algorithms designed to be performed in fixed-point (Angoletta).

If needed, it is possible to perform floating-point arithmetic on a fixed-point processor through software routines that emulate the behavior of a floating-point device. The software routines may consume an extreme amount of clock cycles so in practice it is advised to proceed with caution. Fixed-point DSPs can also employ block floating-point, where a group of numbers with mantissas but a similar exponent are processed as a block of data. Block floating-point is handled in software with the assistance of hardware features ("Choosing a DSP Processor").

Data width will have a major impact on the cost of DSP chosen. The width will influence the size, number of package pins, and the size of external memory devices connected to the DSP. A good design practice is to see what minimum width each of the peripherals will require, as well as, the amount of bits required to represent the signal without losing precision. It is important to note that many, but not all, DSP processors will use an instruction word size equal to their data word size ("Choosing a DSP Processor").

3.3.2 Speed and Memory

There are a number of ways to test a processors speed but first consider the instruction cycle time. The instruction cycle time is the amount of time required to execute the slowest instruction on the processor. If the reciprocal of the instruction cycle time is taken, divided by one million and multiplied by the number of instructions executed per cycle then we will arrive at the processor's instruction execution rate in millions of instructions per second, MIPS. When comparing DSPs MIPS rate, check the DSPs hardware because DSP chips may have phase-locked loops (PLLs) that allow the use of a lower-frequency external clock to generate the needed high-frequency clock on-chip ("Choosing a DSP Processor").

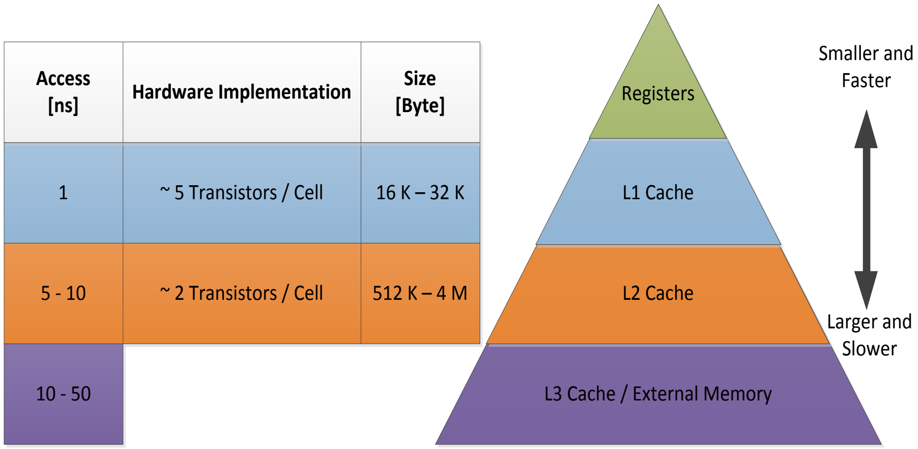
Fast execution of MAC operations requires fetching an instruction word and two data words from memory every clock cycle. To permit this, the DSP will need to be able to access memory multiple times for each instruction cycle, separate instruction and data memories, and instruction caches ("Choosing a DSP Processor").

Digital signal processors for real-time processing will need to access data and access it very fast. DSPs need to transfer data to and from memory or other peripherals at a fast rate. Some of the items that determine a fast data access are the type of high-bandwidth memory architectures, specialized addressing modes, and direct memory access. Typically DSPs adopt the Harvard or Super-Harvard Architecture in which there are separate memories for data and program instructions and two separate buses to connect each of them to the DSP core. To improve the Harvard architecture it is recommended to add a bank of fast memory for the program instructions and data cache (Angoletta).

Instruction pipelining has become extremely important component to DSP performance. Pipelining divides the execution of instructions into multiple stages and executes the instructions in parallel. The instructions can essentially be broken up into 3 stages: instruction fetch, decode, and execute. Fetch is when a DSP calculates the address of the next instruction to execute and retrieves the op-code. Decode will route the op-code from fetch and send it to the proper functional unit in which it will execute the instruction and write he results back to registers (Angoletta).

One of the most often cited features of a Digital Signal Processor is its ability to perform one or more multiply-accumulate operations (MACs) in a single instruction cycle. The MAC operation is a useful algorithm that computes a vector dot product, such as a digital filter, correlation, and/or a Fourier Transform. DSPs now are able to incorporate multiple MAC units operating in parallel fashion to achieve more computations per instruction cycle. With multiple MACs, the DSP may run into the issue of overflow. To counteract overflow possibility the DSP will implement “guard” bits to protect the data integrity (“Choosing a DSP Processor”).

The typical DSP hierarchical memory architecture and typical number of access clock cycles, hardware implementation, and size of different memory types can be located in the Figure below provided by Texas Instruments.



Access [ns]	Hardware Implementation	Size [Byte]
1	~ 5 Transistors / Cell	16 K – 32 K
5 - 10	~ 2 Transistors / Cell	512 K – 4 M
10 - 50		

Figure 3 – 1: DSP Memory Architecture

Preferably our DSP will have a large amount of L1 and L2 caches to decrease the amount of time needed to retrieve data to increase our devices ability to process in real-time. Our device will also utilize DDR SDRAM to store operating system resources, the GUI, Flashback algorithm and the recorded programs specified by Flashback users.

3.3.3 Peripherals

In order for our DSP to communicate with other components within Flashback we must have a wide variety of peripherals. We will be looking at devices that contain at least 10/100 Mbps Ethernet, USB 2.0, audio, and video port.

The DSP requires a reasonable amount of flexibility in I/O pins with programmable interrupts and event generation modes to be multiplexed with other peripherals. These I/O pins will be used for additional features to Flashback device.

3.3.3.1 Universal Serial Bus

Universal Serial Bus (USB) is an external bus standard that supports data transfer and is capable of supporting up to 127 peripheral devices. Inside a USB cable there are 4 wires, one for 5 Volts, ground, and two for data. An example of a USB cable is shown below in Figure 3 – 2.

USB cables are able to be run over 5 meters without signal loss (Brain). A newer standard of USB and one that our signal processors will most likely support is, USB 2.0. USB 2.0 is known as the hi-speed USB and is capable of supporting a transfer rate of up to 480 Mbps (“USB”).

USB connectors allow the user to attach mice, printers, external hard drives, and other accessories to your device quickly and easily. The operating system for our processor package must and most likely will support at least USB 2.0. Upon performing enumeration the host device will find out from each peripheral what type of data transfer it wants to perform. Some different types of data transfer are an interrupt, bulk, and isochronous. Isochronous refers to a streaming device such as speakers. Isochronous data transfer caters to our project as many of the items we will do are in real-time streaming of information (Brain).

A USB divides the available bandwidth into frames and the host can control the frames. Frames will contain 1,500 bytes and a new frame will be received every millisecond (Brain).

3.3.3.2 Ethernet

Ethernet is a communications protocol embedded in software and hardware devices to establish a local area network (LAN) connection (Kanye and Wynn). Ethernet cables are very efficient in sending data over long lines of communication; they can reach in the tens of kilometers in distance (Pidgeon). An Ethernet cable is shown below in Figure 3 – 3.

Ethernet follows some simple rules for its operation. Ethernet devices must be able to attach to a shared medium (segment) that will provide a signal flow path which will have devices connected to the segment (nodes). The nodes will communicate in short messages called frames (Pidgeon).

The carrier-sense multiple access with collision detection (CSMA/CD) describes how an Ethernet cable will control the communication among all the nodes. Carrier sense essentially means that before a second transmission can go through the cable, the first data set must be completed. There is also multiple accesses with carrier sense. If for some reason, data is transmitted at the same time, a “collision” will occur. Sometimes an ethernet segment will be called a collision domain because on this domain two different stations cannot transmit at the same time without colliding into one another. If a collision is detected then the transmission will halt and wait a certain amount of time until it decides to attempt to transmit again (Pidgeon).

3.3.3.3 Inter-Integrated Circuit Bus

Inter-integrated circuit bus is more commonly known as $\mathrm{I}^2\mathrm{C}$ bus. This bus is a two-wire serial bus and there is no need for chip select or arbitration logic which makes this bus cheap and simple to implement in hardware. The two signals are serial data (SDA) and serial clock (SCL). $\mathrm{I}^2\mathrm{C}$ bus provides good support for communication with different, slow, on-board peripheral devices. These devices are accessed in intervals and have a low-bandwidth, short distance protocol. The typical $\mathrm{I}^2\mathrm{C}$ operates at speeds of up to 400 Kbps and can go up to the megahertz range. The $\mathrm{I}^2\mathrm{C}$ protocol has a built-in addressing scheme. $\mathrm{I}^2\mathrm{C}$ is used for EEPROMs, thermal sensors, real-time clocks and most importantly as a control interface to signal processing devices with separate, application specific data interfaces. $\mathrm{I}^2\mathrm{C}$ multimedia applications extend from RF tuners, video decoders and encoders, and audio processors (Kalinsky D. and Kalinsky R.). For Flashback, we could use $\mathrm{I}^2\mathrm{C}$ bus to send our data from the video tuners to the video decoders, then to the audio processor after passing through our system-on-chip (SoC).

One of $\mathrm{I}^2\mathrm{C}$’s weaknesses is its ability to send signals over long distances. $\mathrm{I}^2\mathrm{C}$ will work great on a single board and even if it is sent across multiple boards but it cannot work well past a few boards.

3.3.3.4 Multichannel Audio Serial Port

A multichannel audio serial port (McASP) functions as a general-purpose audio serial port that is optimized for the needs of multichannel audio applications. A McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and inter – component digital audio interface transmission (DIT). The McASP module contains sixteen serializers which can be enabled to either transmit or receive. While in the transmit configuration the McASP can send signals to multi-channel digital to analog converters which will then be sent to amplifiers for audio output. The McASP can also be enabled for receive mode. While in this mode the McASP will act as a digital encoder or processor before it is able to send the signals to amplifiers (“TMS320C6000 DSP Multichannel Audio Serial Port (McASP)”).

3.3.4 Video

The DSP must contain at least two video capture channels. Both of these signals will pass through the DSP and be sent to the storage device simultaneously. The primary and secondary signals will need to pass through the DSP so that the System on Chip (SoC) is able to control which MPEG transport stream to decode and display on the television screen. If Flashback is enabled and the user has wandered to a secondary program while the primary program is analyzed then the secondary program will be displayed on the television. If Flashback is disabled then our device can merely display whatever television program the user wishes to view. The two video capture channels need to be at least standard – definition with the capability of high-definition. After our group successfully performs the Flashback concept with standard definition, we will progress to working with high-definition. Due to this sequence, we need a DSP package that is capable of high-definition video capture channels.

3.3.5 Power

Digital Signal Processors are now being offered at reduced voltages such as 3.3, 2.5, or 1.8 V and all of these can operate at the same clock rate as the 5-V version. By reducing the operating voltage, our Flashback device will be able to minimize power and heat dissipation (“Choosing a DSP Processor”).

DSPs have also begun to offer feature modes to reduce power consumption and increase the user’s ability to manage power. One of those features is a “sleep” or “idle” mode. In this feature the user will be able to turn off the processor’s clock to all but certain sections of the processor. To bring the processor back from “sleep” mode an interrupt can awaken the areas needed. In addition, DSPs could come with programmable clock dividers to allow the processor’s clock frequency to be varied under software control in order to use the minimum clock speed required for a particular task. Lastly, some devices are able to disable peripherals that are not in use.

To optimize the efficiency of the Flashback device, it is preferred to have as many similar low-voltage rails as possible. This will first and foremost reduce the amount of power consumed by the DSP.

The second purpose of this is to reduce the amount of heat within the system. Most components are rated for 90 degree Celsius or higher the high temperatures will degrade components quicker than at ambient temperature. Furthermore, if we have many similar voltage rails then we can reduce the amount of components needed to step-down and clean up power supply signals.

3.4 SATA Controller

Serial Advanced Technology Attachment (Serial ATA or SATA) is an attachment that enables mass storage devices to communicate with a motherboard over a high-speed serial connection. These mass storage drives could be solid state drives, hard disk drives, optical drives, etc. The SATA controller has different modes for to determine how the storage device will communicate. The controller can be set to operate in one of the following modes: integrated development environment (IDE), advanced host controller interface (AHCI), and redundant array of independent disks (RAID) ("About SATA Hard Drives and Controller Modes")

IDE mode will set the storage device to run as an IDE or a Parallel ATA drive. While in IDE mode, the drive will perform slower than the others, but it will provide better compatibility when trying to communicate with older hardware ("About SATA Hard Drives and Controller Modes").

AHCI will enable the advanced features on SATA drives, such as hot swapping and Native Command Queuing (NCQ). Hot swap is the replacement of a hard drive, CD-ROM drive, power supply, or other device with a similar device, while the computer system using it remains in operation. This essentially will give the appearance to the input / output controller that the device is still attached even though it is actually being removed and replaced ("Hot Swap"). NCQ is a technology that has been designed to improve the performance and reliability as the transactional workload increases on SATA hard disk drives. If there are multiple commands sent to the SATA drive, NCQ will group the commands in the order of processing efficiency. This will minimize the mechanical workload and increase the performance on the drive ("Native Command Queuing"). The use of AHCI will increase speed relative to IDE.

RAID enables the device to store the same data in different places on multiple storage devices. This will allow input/output operations to overlap in a balanced way and will therefore improve performance. Whenever a system uses multiple drives, the system will have an increased mean time between failures (MTBF) so storing data redundantly will increase the tolerance to failure ("RAID (redundant Array of Independent Disks)"). While in RAID mode, the SATA controller will also enable AHCI ("About SATA Hard Drives and Controller Modes")

3.5 Digital Image Processing

The interest in digital image processing stems from two main applications, improvement of pictorial information for human interpretation and processing image for autonomous machine. People want to understand what is really going on in a picture (“Fundamentals of Digital Image Processing”). For instance, fire control systems from defense contractors use image processing to determine if people or vehicles are moving in a threatening pattern or if they are carrying a threat. Flashback will focus on the second application. Our device will analyze the incoming video stream provided by a cable company and will search for commercials in a real-time fashion.

Images are digitized in spatial coordinates and in brightness for the image to be considered a digital image. A digital image is a 2-D array of light intensity, and the spatial coordinates of the pixel (an element in the digital array). The first step in image processing is to acquire the digital image. Next, the image will be preprocessed to improve the image to increase the efficiency of image processing algorithms. The image must now be segmented into parts or objects. After segmentation, the image will be converted to a form suitable for computer processing. Then the system can extract features that result in information of interest or features that are basic for differentiating objects from another. Based upon the information, the system will assign a label to the objects. Lastly, the image will be interpreted (“Fundamentals of Digital Image Processing”).

3.6 Object Isolation

In comparison with a single still image, video sequences can provide more information about how objects and scenarios change over time. To detect an object’s behavior, the system must first perform an isolation algorithm. The isolation algorithm will start with the current frame and a reference frame. Then the wavelet transform will be applied to each frame to create a wavelet pyramid and the filter banks. The number of levels of decomposition will be chosen depending on the size of the frame and content of the video. The background differences play a large factor in the algorithms ability to track an object.

Therefore, the algorithm will take the current image and the reference image and subtract to find the differences in the image. By doing this, all of the pixels in the frame that are consistent frame to frame will be set to zero and the differences will have a value other than zero. The differences in the images could be a result of either noise, or that an object has moved from frame to frame. To make sure that noise isn’t a factor a threshold must be established for the difference of pixel values. After isolating the object(s), the system must only perform complications on a smaller range of pixel values to track the object (Wang and Doherty).

In terms of our groups project object isolation could be performed to key in on the broadcast logo. As you may have noticed, on television programs the broadcast logo will be displayed in, most likely, the bottom right quadrant of the video stream. The logo will “disappear” when the broadcast station transitions to a commercial break. Our system could employ the object isolation algorithm and detect when that object is no longer there. This would mean, with high accuracy, that the television program is on a commercial break. Then as the logo reappears, our detection algorithms will trigger our system-on-chip to prompt the user and let the user know that the commercial break has ended.

3.7 Real-Time Processing

Flashback will need to operate in real-time. What this means is that our processor must perform calculations on the first set of data before the second set of data arrives. Digital signals are represented with sets of samples of an analog signal at discrete moments in time. Therefore the Digital Signal Processors ability to process signals in real-time depends on the sampling rate. The sampling frequency must be at least twice the frequency of the highest frequency component of interest in the signal. For example, if we only look at a 10 kHz input signal then our sampling frequency must be at least 20 kHz. To process this data in real-time the signal processor must be able to compute in the following time budget:

$$\text{time budget} = \frac{1}{20\text{ kHz}} = \ 50\ \mu\text{s}$$

To enhance Flashback devices ability to process in real-time we will utilize the following guidelines. First, work in parallel as much as possible to increase computation speed. Working in parallel avoids significant waiting times for executing operations in comparison to working in series. Second, specify and fully document all interfaces. Third, always include validity checks on DSP inputs. Fourth, add spare parameters with different formats to maximize flexibility, as well as, allowing the addition of debugging features or implementation of small updates. Finally, validate code implemented on device.

3.8 MPEG

Moving Pictures Experts Group (MPEG) is a standard for “the generic coding of moving pictures and associated audio information.” It describes a combination of video compression and audio data compression methods which permit storage and transmission of movies using available storage media and transmission bandwidth. MPEG is widely used as the format of digital television that are broadcasted over the air, cable, and satellite systems ("MPEG-2 Video Encoding (H.262)").

MPEG video encoding (H.262) employs an intra-frame Discrete Cosine Transform (DCT) coding and motion-compensated inter-frame prediction. A DCT expresses a finite sequence of data in terms of a sum of cosine functions oscillating at different frequencies. The DCT breaks up each frame into 8 pixels by 8 lines of each component of the picture. Then a DCT is performed on that to produce blocks of DCT coefficients. The magnitude of these coefficients indicates the contribution of a particular combination of horizontal and vertical spatial frequencies to the original picture block. This is very important lossy compression of audio and images.

Next, MPEG employs a motion-compensated interframe prediction. This prediction looks at the current frame and compares it with a reference frame (previous frame) to determine how many picture elements have changed. After finding the redundancies, the MPEG encoder will eliminate the elements from the picture. Typically the redundancy calculations are done prior to DCT in order to avoid duplicate / excessive calculations ("MPEG-2 Video Encoding (H.262)").

MPEG encoding can be done purely in software or by using an MPEG capture card or a video editing card with in-built MPEG encoding capabilities, or a dedicated hardware encoder like the ones typically built into video recorders. The transport stream contains compressed images, sequence headers, group-of-picture headers, and other data needed to decode the transport stream. The sequence header contains information concerning the frame size and the frame rate needed before the system can decode the stream ("MPEG-2 Video Encoding (H.262)").

MPEG-2 defines 3 different types of pictures: Intra pictures (I-frames), Predictive pictures (P-frames), and Bidirectional-predictive pictures (B-frames). I-frames are coded without any reference to other pictures. P-frames use motion compensation as described above and could actually be used as a reference frame for further predictions. B-frames can use previous and next I-frame or P-frame for motion compensation. In addition, B-frames will offer the highest degree of compression.

Before a future image is able to be used as a reference frame the encoder must reorder the pictures from natural ‘display’ order to ‘bit stream’ order so that the B-picture is transmitted after the previous and next pictures it references. B-frame technique could come in handy with Flashback algorithms and processing in real-time. Our system desires to do a comparison of frames because when broadcast stations transition to commercials they insert a frame where all pixel RGB values are black. Since the entire frame is black all elements in the DCT blocks will be changed and transmitted. If our system can look ahead one frame, this will ease our system’s ability to process in real – time ("MPEG-2 Video Encoding (H.262)").

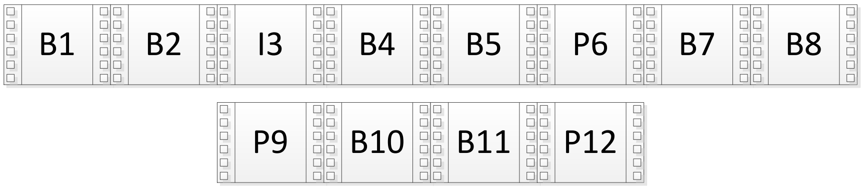


Figure 3 – 2a: Group of Pictures in (a) 'display' order

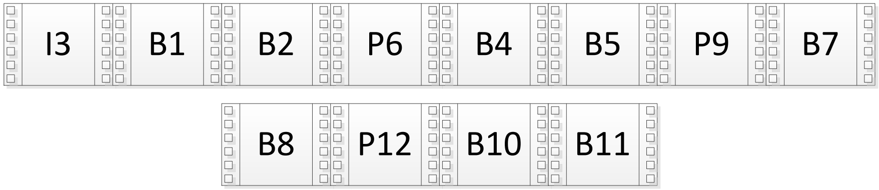


Figure 3 – 2b: Group of Pictures in 'bit stream' order

In the video editing and processing systems, multiplexing refers to the process of interleaving audio and video into one coherent MPEG transport stream. The individual audio and video streams may have a variable bit rate. Recording the digital signals provided by the MPEG stream is straightforward since no additional hardware will be needed to quantize and compress the signal ("MPEG-2 Video Encoding (H.262)").

3.8.1 H.264

Transform coding for H.264, also known as MPEG – 4, is a more efficient image compression technique than MPEG – 2 H.262. H.264 first transforms the image into blocks of four elements by four elements as opposed to the eight by eight (DCT) in MPEG – 2. Next, MPEG – 2 was a floating-point arithmetic compression and H.264 is an integer to integer transform. The integer transform allows for exact reconstruction and completely eliminates decoder drift caused by a mismatch from encoder/decoder. After getting the elements in four by four blocks, there are nine different ways that H.264 can perform spatial prediction. The inverse transform will now only require 16 bits of integer precision. Furthermore, H.264 can have multiple reference frames of up to 15 frames and can also refer to the future frame by the means fo displayed the frames in 'bit stream' order (Igarta).

3.9 Video Tuners

Flashback provides the ability to record and display one video channel while also record and analyze a second channel. For this to be possible, Flashback will require dual video tuners. Video tuners tune into a particular television signal as specified by the user. The uses of dual tuners allow the capability of recording a live program while watching another live program simultaneously. There are many different types of video tuner ICs. Some of the tuners are geared toward digital only, analog only, or some are a hybrid of digital and analog signals.

Analog tuners can support three different types of analog television systems; National Television System Committee (NTSC), Phase Alternating Line (PAL), and Sequentiel Couleur avec Mémoire, which translates to Sequential Color with Memory (SECAM). NTSC is an analog television system that is used in most North America, parts of South America, and some Asian countries. There are 30 frames per second (fps) transmitted each second and these frames are composed of 525 individual scan lines. PAL is mostly used overseas in Middle Eastern, parts of South America, Europe, and Asia. PAL uses 25 fps and each frame is composed of 625 scan lines. SECAM uses frequency modulation to encode chrominance information. SECAM takes the post-production analog signal that was done in PAL and then transforms it to SECAM at the point of transmission ("PAL & NTSC & SECAM").

Digital tuners support Advanced Television Systems Committee (ATSC) / Quadrature Amplitude Modulation (QAM), Digital Video Broadcasting – Terrestrial and -Cable (DVB-T or DVB-C) or various newer standards (DVB-T2/C2), Integrated Services Digital Broadcasting – Terrestrial and -Cable (ISDB-T/C), and lastly Digital Terrestrial Multimedia Broadcast (DTMB). ATSC employs MPEG-2 data compression standard to perform a 50:1 reduction in data (Nist). QAM is both an analog and digital mod scheme; QAM changes the phase and amplitude of carrier waves using either a Amplitude-Shift Keying (ASK) or Amplitude Modulation (AM) ("Quadrature Amplitude Modulation (QAM)"). DVB-T2/T is the standard for transmission of digital television signals from aerial antenna to aerial antenna ("What Is DVB-T?"). DVB-C2/C is the digital broadcast standard that uses cable as the transmission medium. ISDB is a Japanese and Brazilian version of DVB with improvements in mobile reception ("Comparison of 3 DTTB Systems"). Lastly, DTMB is the Chinese version of DVB-T utilizing multiple carrier signals to transmit television content ("DTMB Technology").

Hybrid tuners are able to capture both analog and digital television transmissions. Tuners can be selected for low power consumption and operate on 3.3 V and 1.8 V power supplies with a typical dissipation of only 1 W. They also feature a low-noise amplifier (LNA) and high-Q tracking filter which will provide gain only around the channel desired (Schweber).

3.9.1 PLL Frequency Synthesizer

A frequency synthesizer is important in any transceiver. Frequency synthesizers must be able to look at different frequency bands and for different wireless standards. A Phase-Locked Loop (PLL) circuit will synchronize the output signal with an input reference signal. This means that the two signals will have the same frequency and will operate at a constant phase difference. PLLs in television applications perform the horizontal and vertical synchronization and color reconstruction. PLL can be used as a frequency synthesizer by placing a loop divider in the feedback path as shown below:

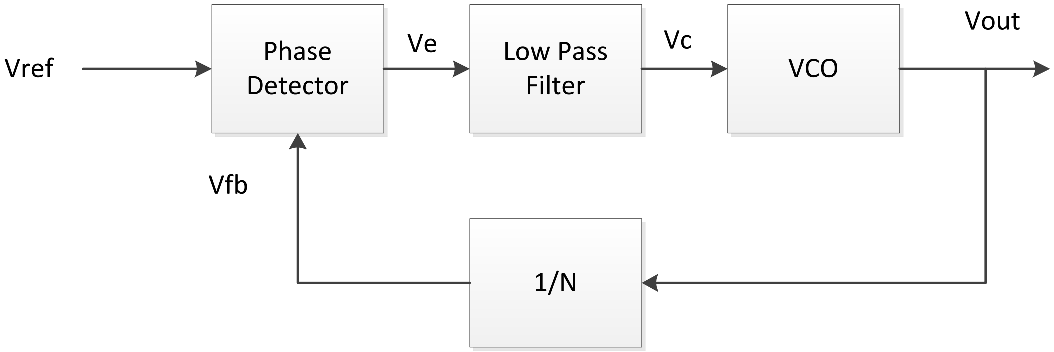


Figure 3 – 3: Phase-Locked Loop Frequency Synthesizer

The output frequency for the diagram above is equal to where is the Voltage Controlled Oscillator (VCO) center frequency and is the VCO conversion gain. By adjusting N we can adjust the frequency of the PLL synthesizer. In order to allow the frequency synthesizer to account for a larger range of frequencies, the VCO's conversion gain must be very large. This might bring on some unwanted phase noise so the diagram should implement a switched tuning VCO instead. The switched VCO can tune over a wide range while still maintaining a low conversion gain (Palermo).

The use of a frequency synthesizer in the Flashback device could help hone in on certain video signals. By controlling the parameters of VCO and the feedback loop, we could adjust the PLL to accommodate for all frequencies within the over-the-air television spectrum.

3.10 Grounding in Mixed Signal Systems

Maintaining a strong, wide dynamic range analog signal with minimal noise next to a digital environment will depend upon signal routing, decoupling, and grounding. Mixed-signal integrated circuits could have low or high digital currents. The different levels of digital current will need to have different grounding schemes to obtain optimum performance. It is desirable to separate sensitive analog components from the noisy digital components and hence the ground planes for analog and digital circuitry (Kester and Bryant).

A low impedance large area ground plane is critical to all analog circuits. This ground plane will act as a low impedance return path for decoupling high frequency currents (caused by fast digital logic) and it will also minimize EMI/RFI (electromagnetic interference / radio frequency interferance) emissions. All integrated circuit ground pins should be soldered directly to the ground plane to minimize inductance and resistance. Power supply pins will need to be decoupled to the ground plane using low inductance capacitors (Kester and Bryant).

At least one entire layer of a PCB system should have a dedicated ground plane. If it is a double-sided board then one side should be ground and the other should be interconnects to the best of the designer's ability. Due to power, interconnects, and through holes, reserving one side of the board for ground will be near impossible. So best practice is to keep at least 75% of the board area on one side for ground plane. In addition, use at least 30% to 40% of PCB connector pins for ground. Lastly continue the ground plane on the backplane motherboard to power supply return (Kester and Bryant).

The analog to digital converters (ADCs) and digital to analog converters (DACs) and other mixed signal ICs with low digital current should be treated as an analog component and grounded and decoupled to the analog ground plane (Kester and Bryant).

After designating an analog and a digital ground plane, it is recommended to tie these planes together through a chunk of metal. This will allow the noisy digital currents to flow through the power supply and digital ground plane then back to the digital supply. The analog plane will not be affected (Kester and Bryant).

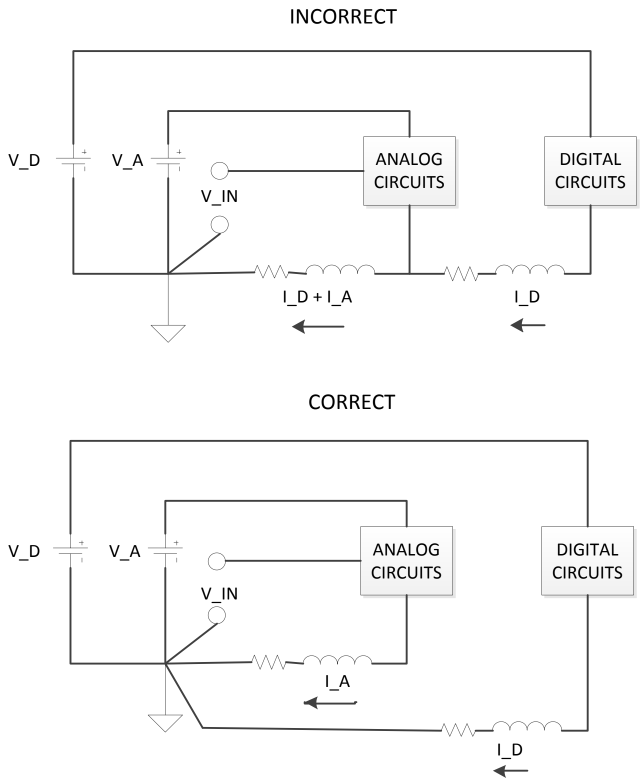


Figure 3 – 4: Proper Grounding Scheme

3.11 DDRx SDRAM

DDRx SDRAM stands for Double Data Rate Type x Synchronous Dynamic Random Access Memory; “x” is the number of revision to the DDR standard. With each new revision, DDR SDRAM is able to transmit at a higher theoretical transfer rate and the operating voltage lowers. SDRAM means that they use a clock signal to synchronize data. Dynamic RAM means that data is stored in an array of capacitors. DDR memory transfers two data sets for each clock cycle. Because DDR SDRAM sends two data sets per clock cycle, they are able to achieve double the performance of memories without this feature running at the same clock rate. DDR utilizes the rising and falling edges of clock cycles to transmit information (Torres).

Usually DDR, DDR2, and DDR3 operate at typical voltages of 2.5 V, 1.8 V, and 1.5 V respectively. Each revision, in addition to transfer rates, had a goal of reducing the total power consumption. If the memory is being overclocked then the operating voltage can raise approximately ten percent. The power consumption is a factor of speed, type of usage, and operating voltage (Torres).

DDR3 memories have the highest latency. Latency is the time the memory controller has to wait between requesting the data and sending the data. Even though DDR3 has the highest latency value, it is based on the number of clock cycles passed before receiving or sending data. This means that the latency value doesn’t necessarily correlate to a longer wait time (Torres).

DDR memories also feature On-Die Termination (ODT). ODT is a resistive termination network located inside the memory chips to provide cleaner signals. ODT also increases the time frame for memory to read or write data which will allow better clocks to be achieved (Torres).

3.12 Thermistor vs. Analog Temperature Sensor

Thermistors and analog temperature sensors are the two most popular sensors for gauging temperature for electronics and their applications. The thermistor will require a resistive network to linearize the relationship between output voltage and temperature. While operating in room temperature range, the thermistor shows an almost linear slope and can have the “linear” range altered by the resistor network. However, once the system gets too hot or too cold, the thermistors will no longer have a linear relationship and the actual temperature value reading from the thermistor will not be accurate.

An analog temperature integrated circuit can output an exact linear response over the temperature range -50 degrees Celsius to 150 degrees Celsius. In the terms of supply current versus temperature, the greater the temperature the greater the supply current desired for the thermistor. The thermistor supply current range could be from single digit micro-Amps to a few hundred micro-

Amps. As temperature varies for analog temperature integrated circuits the current could fluctuate by about four to five micro-Amps (Gosselin).

3.13 Debugging

A starting point of debugging the real-time design flow is to make sure that our group has an executable code; one that doesn't have compilation and linker errors. We will use this code to make sure the code behaves as expected. This could be something as simple as switching on and off an LED (Light Emitting Diode). To debug code, our DSP will use the following tools: simulation, emulation, and real-time debugging techniques. Simulation tools will provide full visibility into the DSPs internal registers, peripheral performance, and interrupts. Emulation tools will embed debug components into the target to all an information flow between Flashback device and host computer. Lastly, real-time debugging techniques allow a real-time data exchange between host and device without stopping the DSP (Angoletta).

3.14 ADC and DAC

An analog signal means that the signal is continuous over time. Real world signals such as sound, light, heat, etc. are all analog signals and these signals can be shown through electrical signals. A digital signal is the sampled version of the analog signal.

To convert analog real-world signals to a digital form that can be manipulated by a computer our group will use an analog to digital converter (ADC). When digitizing an analog signal, the waveform is discretized and quantized. Discrete samples are dependent upon the frequency of the sampler. If the sampler is run at 1000 Hz then the analog signal will be broken into a sample every 1/(1000 Hz) or every millisecond. Ideally, a system would have an extremely high sampling rate (infinity or close to it) but this will also require more storage capacity. How do you know the best sampling rate? By looking at the Nyquist Rate we can determine that our desired sampling rate will be at least twice the maximum frequency to avoid aliasing effects.

Quantization breaks up the amplitude of an analog signal into change in voltage divided by the amount of levels in the converter. Quantization depends upon the N number of resolution bits for the converter. If a system has a 12 bit resolution then the total number of quantized levels are equal to: . If the resolution is poor then there will be significant quantization noise and rendering the original signal will be impossible (Torres).

Since digital signals are represented by only two numbers, noise can easily be disregarded from the original signal (assuming proper resolution ADC). When playing back the digital signal the noise you may hear is almost without a doubt from the system components and not the digital signal. Analog signals are extremely vulnerable to noise and it’s nearly impossible to extract the original signal if another signal is operating in the same frequency range. In addition to eliminating noise, digital signals also have compression capability. Compression will help shrink down the file size and save space on the storage device or in the bandwidth (Torres).

To play the audio and video for the user our system will need to convert our modified digital signal back to analog via a digital to analog converter (DAC). DACs convert the binary signal back to voltages. This will not provide an exact depiction of the original input signal due to the resolution of converters but it will be negligible if ADC and DAC have good resolution for application (Torres).

3.15 SPDIF

Sony/Philips Digital Interface (SPDIF) is a data link layer protocol and a physical layer specification for transmitting digital audio signals over optical or electrical cable. This means that the information is transmitted in 0’s and 1’s instead of an analog format. The audio will have a higher fidelity because no noise will be added to the audio signal since the signal is transmitted digitally. The difference between SPDIF and HDMI (High-Definition Multimedia Interface) is that SPDIF only transmits audio. SPDIF should be used when trying to interface with audio equipment, such as surround sound systems. There are two kinds of SPDIF connectors: a coaxial cable (RCA) and an optical cable (Toslink). Between the RCA and Toslink, there isn’t much difference between audio qualities but optical cables are less sturdy than coaxial. If someone were to bend or step on the Toslink cable then it could lead to permanent damage (Torres).

3.16 Hard Disk Drives

Hard Disk Drives, which are otherwise known as Hard Drives, are a type of storage devices used in a wide range of computers. They are also known as a backing store, basically a computer system’s main memory, where important data is stored for future or continuous access.

A hard disk drive as we know them today, consist of single to multiple magnetic disks, otherwise known as platters. These disks are aligned on a spindle that will spin them at a set speed so that they may be read from using multiple read and write arms .

These disks have multiple tracks and within these tracks there are sectors. Each disk is addressed using logical blocks. They can be thought of as a one – dimensional array of logical blocks. These blocks are sequentially mapped into the aforementioned sectors. With multiple disks each track is a part of a cylinder.

The mapping of sectors begins at the first sector of the first track on the outermost cylinder.

These kinds of mapping read and write speeds of a hard disk drive are dependent upon how fast the read/write arm's motor can move to the requested track. Then the speed is dependent upon how fast the spindle containing the disks can rotate the disks to the requested sector. Common drive speeds are 5400 Rotations Per Minute (RPM) and 7200 RPM. Drives with a rotation speed of 7200 RPM are usually more expensive being that they are faster.

Hard drive technology has evolved greatly, and the biggest achievement our group has found is the size. The available form factors for hard disk drives are 3.5'' and 2.5''. If our group decides to use a hard disk drive for storage it will be of 2.5'' form factor.

The Pros of Hard Disks are:

- Portability – Hard Disk Drive manufacturers design hard disks with a number of input/output interfaces. These interfaces include Serial Advanced Technology Attachment (SATA) and the slower Integrated Drive Electronics (IDE or Parallel ATA).
- Built in device controller – Hard Disks have their own device controller so they can easily be installed into motherboards that support the disk's I/O interface.
- Price – Because hard disk technology has existed for years and with the advance in technology that spawned the personal computer, there are many kinds of hard disks in existence today. This advancement in technology, along with other factors, drives the manufacturing cost down.
-

The cons of Hard Disks are:

- Seek time can be slow – Seek time is the time it takes to move the disk's read/write head to a cylinder.

Durability – Hard disks are subject to damage caused by extreme physical shock, such as falling to the ground. Because there are multiple disks spinning only a few millimeters away from the read/write head, a sudden drop could cause the head and the disks to make contact and damage the magnetic surface of the disk.

3.17 Solid State Drives

Solid State Drives are a form of electronic storage, just like a hard disk drive it acts as the secondary storage for a computer system. They use the concept of Solid State Storage (SSS) which can be found in flash drives. Unlike their hard disk counterpart, solid state drives have no moving parts. They achieve data persistence through the use of integrated circuits. These circuits consist of NAND based flash memory which will retain data without power, hence data persistence is achieved.

Much like hard disk drives, solid state drives use common Input/Output (I/O) interfaces such as Serial ATA. Solid state drives are becoming more popular for a few key features:

Resistance to physical shock – Because solid state drives have no moving parts they are less likely to damage themselves when dropped unlike their hard disk counterparts.

Onboard device controller – Solid state drives are built to support plug and play, just like hard disks. The onboard controller allows the drive to easily interface with a mother board that supports the same I/O interface that the drive uses.

Low access time – Because solid state drives have no moving parts and uses NAND flash memory, access times are a 100 times faster than a hard disk drive.

Fast read speeds – Using various improvements upon the SATA interface called SATA I, II, and III, more information can be read and written to the drive a lot faster. These read and write speeds can be physically limited by the drive architecture, e.g. a 5200 RPM Hard Disk Drive.

These features are the reason that solid state drives are being advertised as hard disk replacements. While Solid state drives come with many positive features that make them better than a Hard Disk Drive in almost every way, they do come with some unfortunate characteristics. The biggest drawback to Solid state drives is their short life span, which is affected by how much the drive will be written to bring.

Since we are attempting to accomplish real time playback, fast read speeds are essential. Excessive television lag defeats the purpose of what Flashback is trying to bring about. The read and write can possibly be expanded if we create our own kernel.

Just as any hard disk drive that is used daily will ware, so will a solid state drive. The only difference is that a solid state drive will wear much faster than a hard disk drive due to the characteristics of NAND flash memory. Each bit in memory can only be written to, a variable number of times, before its value can’t be changed any more (Lambert). This variable is the basis of most solid state drive life span estimations. Generally it is believed that a bit in solid state drive memory can survive up to 10,000 writes (Lambert). This number leads some to believe a solid state drive can last up to five years while some believe they will last about one to two years (Lambert). Most solid state drive architects have created a number of techniques to combat the wearing of each bit and prolong the life of a drive. Some of these techniques are briefly described in Table 3 – 2.

Features	Description
Wear Leveling	The disk controller keeps track of how many erase cycles have been performed on each block of memory. It will then remap the logical blocks to physical blocks with lower erase cycles. This spreads out the use of all available blocks and thus spreads the wear out over the drive.
Bad Block Mapping	The disk controller will keep track of which blocks are going bad or ones that have gone bad (inaccessible or read-only). It will then remap the logical blocks to the working physical blocks and will no longer try to use these bad blocks for storage.
Read and Write Caching	Read and Write caching is a process used by the disk the controller to speed up read and write speeds. Data is stored in a cache, typically RAM or the disk's cache for some time, until it is actually written to disk or is not being frequently accessed anymore.
Error Correction	Error correction is a process used by almost all modern day storage devices. The disk controller uses various processes such as error correcting hamming codes and parity bits in formulas to recalculate an expected sum after data transmission or write. These codes and formula can be used to present a user with the correct data after an error is detected.
Garbage Collection	The disk controller will use this process in the background or between peak periods of writing to maintain normal write speeds. The garbage collection process will accumulate data that has been marked for erasure by the operating system as garbage. The disk controller will then perform a whole block erase in order to reuse the block.

Table 3 – 2: SSD Controller disk endurance enhancement techniques (Kerekes)

The most notable of the highlighted techniques in Table 3 – 2 is wear leveling and garbage collection. Wear leveling is critical to the longevity of the solid state drive, while garbage collection is critical to the drive's performance. Garbage collection as described in Table 3 – 2 can be compared to disk defragmentation in hard disk drives. Disk defragmentation is to connect empty fragmented memory blocks together to form larger blocks of writable memory. Although these features exist in almost all drives, to improve solid state drive technology, there is still a difference in performance and price when it comes down to the two main types of solid state disk drives. These primary types are known as Single Layer Cell (SLC) and Multi-Layer Cell (MLC) architectures.

3.17.1 Single Layer Cell vs. Multi-Layer Cell

This section was sourced from Manek Dubash and Andrew Buss and is only a brief comparison of the two primary types of solid state memory cells.

As discussed previously, all flash memory suffers from wear, which occurs because erasing or programming a block of memory (cell) subjects it to wear due to the voltage applied. When a write to memory occurs, a charge is trapped in the transistor's gate dielectric and causes a permanent shift in the cell's characteristics, which, after a number of write cycles, manifests as a failed cell (Dubash).

Single Layer Cell memory uses a single cell to store one bit of data. Multi – Layer Cell memory is naturally more complex and can interpret four digital states from a signal stored in a single cell. This design makes the circuit board a lot denser for a given area and hence cheaper to produce. It is for this reason that Multi – Layer cells will wear out faster than Single Layer Cells.

As mentioned before, a Multi – Layer Cell typed memory is typically rated at 10,000 erase/write cycles, while a Single Layer Cell typed memory could last 10 times that before failing. However, manufacturers of products consisting of Multi-Layer Cell typed memory can and do have the technologies and techniques, described in Table 3 – 2 at their disposal.

According to Buss, “most such techniques are implemented in the device controller -- the interface between device and computer -- with companies such as SandForce and Intel among the most advanced in implementing such techniques.” (Dubash)

3.18 Software Operating Environment

This section will discuss the potential operating environments Flashback may implement. The environment chosen will limit our choice of processors, since it has to be fully compatible. Reliable and efficient communication between the software and hardware are essential to the success of Flashback.

3.18.1 Operating Systems

An Operating System, as defined by a popular search engine definition is, “the software that supports a computer's basic functions, such as scheduling tasks, executing applications, and controlling peripherals.”

That being said, we decided that since Flashback, at a software level, will be a complicated algorithm and GUI and will require a number of threads to run efficiently, we will use an operating system that supports multi-threading, high level programming languages such as C/C++ and Java, and a sophisticated/mature Kernel. There are a few types of Operating Systems, the ones that we thought were of note and could fit our use case described above are the Real-Time Operating System (RTOS) and the Multi-Tasking Operating System.

A Real-Time Operating System is an Operating System (O.S.) that aims to execute functions and instructions of an application or program in or as close to real-time as possible. A RTOS interfaces tasks with peripherals such as a Direct Memory Access (DMA), Input / Output (I/O), and memory through an Application Program Interface (API) (Angoletta). It achieves this speed by employing the features of efficient scheduling algorithms, whose only purpose is to provide any event that occurs with the necessary system resources to complete its task in the fastest and most predictable way possible. While a Multitasking O.S. uses the concept of preemption to schedule and manage its applications and their threads. This concept is similar to that of the scheduling done by a Real-Time Operating System with the only difference being when programs are executed or allowed to run on the system's processor. A Multitasking operating system will share its available hardware resources between programs in memory and will decide when to give each program use of certain resources.

3.18.2 FreeRTOS

FreeRTOS is an open source Real-Time Operating System that is free for download and commercial use. It is essentially a Real-Time Kernel with support for complex instructions written in C. It is designed to be small enough to fit onto a micro-controller and robust enough to perform tasks prioritized by the developer in a very deterministic and predictable way. This allows for low-level instructions to build modular software that is efficient, powerful and reliable, since tasks are performed only when the developer wants them to be performed.

3.18.3 ArchLinux ARM®

Arch-Linux is an independently developed operating system that is based on the GNU\Linux distribution (Arch Linux). The core ideal of this Operating System is minimalism and simplicity when it comes to the functionality and features. Arch-Linux was built with the idea that the user, a competent and avid user of Linux Kernel distributions, will build and shape the Operating System to meet their needs. Arch-Linux, on installation, provides the user with a command line interface, text editor and package manager to download, install, or develop the necessary software to have the system meet the user's needs. With this approach Arch-Linux only makes use of what the user wants and needs, meaning only what is necessary to the system’s functionality is installed and is running.

A forward-thinking and convenient feature of Arch-Linux is its rolling-release update model for its installed packages and core system files (Arch Linux). Arch-Linux does not need large programs or files to update itself or its packages because packages are installed from the most recent and stable source that has been made available by their developers. In the case of updates, the Arch-Linux system and user installed packages can be installed through small updates instead of large scheduled releases such as in the case of Microsoft Windows or Ubuntu.

Another feature of Arch-Linux is that since it is based upon the GNU\Linux Distribution it can easily be ported, or tailored, for different architectures and the processors that use said architectures. An operating system that we came across that seems mature and powerful is the Arch-Linux Arm Distribution. As defined by the information I found on their web site: It is “a distribution of Linux for ARM computers.” The developers and the community of Arch-Linux ARM “provide targeted kernel and software support for soft-float ARMv5te, and hard-float ARMv6 and ARMv7 instruction sets on a variety of consumer devices and development platforms.” (ArchLinux) Being that Arch Linux ARM is a variant of Arch-Linux it also standbys the philosophy cf “simplicity and user-centrism” and making their target “competent Linux users”. (ArchLinux) Users are given “complete control and responsibility over the system”, which is the main feature that interested us since we will be developing our own GUI in C++. Because Arch-Linux is a variant of a GNU\Linux distribution it natively supports high level languages such as C and C++. We have personally tested and can confirm this support (ArchLinux).

3.19 Software Development Environment

To develop the Flashback algorithm and graphical user interface we will need an Integrated Development Environment (I.D.E.). An Integrated Development Environment is formally defined “a programming environment integrated into a software application that provides a Graphical User Interface builder, a text or code editor, a compiler and/or interpreter, and a debugger” (Webopedia).

3.19.1 Microsoft Visual Studio 2012 Professional

Microsoft Visual Studio 2012 Professional is a version of an Integrated Development Environment that is built to allow developers to develop software that can be Cross-Platform, works across multiple operating systems, or specifically target the Windows platform. As a powerful I.D.E. Microsoft’s Visual Studio supports many programming languages such as:

- C#
- Visual Basic (V.B.)
- JavaScript
- C++
- C
- HTML
- Downloadable plugins
  - Verilog
  - Assembly

Our group is interested in using Microsoft Visual Studio 2012 Professional because it also has support for third – party “add – ins”. This is useful because our group plans to use a cross-platform user interface framework, which supports C++, to build our graphical user interface. The developers of this framework have also developed an “add - -in” for Microsoft Visual Studio 2012. This enables users to develop on a Windows platform and target other platforms such as Linux, in our use case. This “add – in” also does cross compilation support for different processor architectures.

3.19.2 Qt Creator

Qt Creator is another cross-platform Integrated Development Environment which is similar in functionality to Microsoft’s Visual Studio I.D.E.. It was developed by the developers of the aforementioned user interface framework. This framework is named Qt and is discussed in more detail in section 5.1 (Graphical User Interface and Overlay) of this document. As defined on Qt Creator’s website this development environment features (Qt).

- Code editor with C++, QML and ECMAscript support
- Rapid code navigation tools
- Syntax highlighting and code completion
- Static code checking and style hints as you type
- Support for source code refactoring
- Context sensitive help
- Code folding
- Parenthesis matching and parenthesis selection modes

Qt Creator also features a visual debugger for C++ which enables the user/developer to graphically step through the line – by – line execution of their compiled code. This simplifies the process of debugging when compared to text or command line debuggers such GDB in the Linux platform.

One of the major advantages of Qt Creator is that it allows a team of developers, to share a project across different development platforms with a common tool for development and debugging. Qt creator’s documentation has outlined the following features and its documentation can be found online (qt.digia.com).

Project Creation – to set up a project, you first have to decide what kind of an application you want to develop: do you want a user interface based on Qt Quick or HTML5 or Qt widgets. Second, you have to choose the language to implement the application logic: C++ or JavaScript.

Version Control Systems – The documentation's recommended way to set up a project is to use a version control system. Store and edit only project source files and build system configuration files (for example, .pro and .pri files for qmake). It is suggested not to store files generated by the build system or Qt Creator, such as makefiles, .pro.user, and object files.

Project Configuration – Qt Creator allows you to specify separate build settings for each development platform. Special files called shadow builds are used, by default, to keep the build specific files separate from the source. You can create separate versions of project files to keep platform-dependent code separate. You can use qmake scopes to select the file to process depending on which platform qmake is ran on.

Session Management – Items such as open files, breakpoints, and evaluated expressions are stored in sessions. They are not considered to be part of the information shared across platforms.

Qt creator also features two integrated text and visual editors called Qt Quick Designer and Qt Designer. They allow you to create quick Qt projects from scratch or import existing projects to develop applications.

3.20 Signal Input\Communication Mediums

This section will discuss potential signal input mediums. We will focus on those more common towards televisions, since we are trying to implement our device towards that media.

3.20.1 Coaxial Cable

Coaxial Cable, also known as Coax, is a common data or signal transmission medium. There are two kinds of coaxial cables that are widely used. The one we're interested in is the 75-ohm cable, which as stated by Tanenbaum, in Computer Networks 4th Edition, is "commonly used for analog", and nowadays High-Definition media transmission and cable television (Tanenbaum). As you can see in Figure 3 – 9, a coaxial cable consists of a copper wire core, insulating material, a braided outer conductor, and a protective plastic coating.

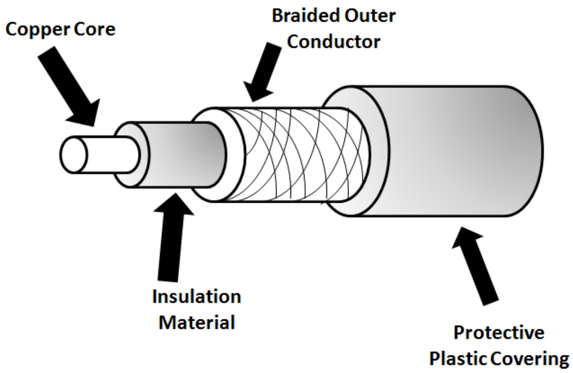


Figure 3 – 5: A detailed cutaway of a Coaxial Cable.

This design is meant to confine the electromagnetic signal that is to be carried by the copper core, to the wire without inducing currents. It also allows for bending of the wire without changing the impedance, which could cause the signal to be reflected back to its source and/ or canceling of the transmission signal. An actual coaxial cable encased in its protective covering with the RG6 connector is shown below in Figure 3 – 10.

3.20.2 Component Video

Component Video is another medium through which video transmissions and signals can be passed on. According to wiseGeek, component video "is one of the high-end video interfaces." It is better than composite video, which is described below in section 3.20.3 and S-Video, and provides greater signal clarity than both of these technologies. Component video is a color format that splits three traditional video signals into their own channels (PCMag). These colors are green, red and blue, and they require three separate cables to be placed on their own channels (wiseGeek).

There are two types of component video, digital and analog. The more prevalent of the two signals is analog component video. It is used mostly in televisions, DVD players, and notably in cable boxes and satellite television receivers (PCMag).

As described by wiseGeek Component video cables are available in different configurations. All three video cables "might be encased in a single sheath with triple tails at each end, or the cables might be bonded or ribboned together." Three single RCA or coaxial cables can also be used, but should be the same length and type. Increased cable quality, tips and shielding add to the cost, as will the length of the cables (wiseGeek). These wires usually come in five wire sets.

3.20.3 Composite Video – RCA

A type of video transmission medium, and analog signal that our group is interested in is, the combination of composite video signal using a Radio Corporations of America’s (RCA) interface. The plug shown in Figure 3 – 13 is the interface referred to as a male RCA plug.

Composite video is a type of signal that contains all the necessary video information for displaying media on a compatible display device. The component video signal is broken down into two signals, a Y signal for luminescence or brightness and a C signal for color (wiseGeek). Brightness amd Contrast levels are provided by the Y signal, so as to control the blacks and whites. The C signal dictates the intensity of the Red, Green and Blue levels (wiseGeek). These signals are then decoded from the single composite video signal by a filter embedded into the hardware of the receiving device.

There are two major drawbacks of composite video, the first being that composite video does not support audio, so the cables are usually partnered with red and white audio cables for stereo sound output. The second drawback is that the highest video quality possible is 480i, which is a video mode widely known as “Standard Definition Television” (SD – TV) and standardized by the National Television System Committee (NTSC) (Pirillo).

As previously mentioned, composite video cables only carry video signals, so the audio has to be delivered through other means. One of them being through composite audio cables which carry the left and right channels of the composite video’s associated audio. One would usually see a composite cable coupled with red and white audio RCA cables.

3.20.4 High Definition Multimedia Interface (HDMI)

High Definition Multimedia Interface, widely known as HDMI, is a video interface and protocol that was designed and released to the consumer market in 2003 by the HDMI forum. The HDMI forum’s Facts and Questions website describes HDMI as the following: “HDMI (High-Definition Multimedia Interface) is the first and only industry-supported, uncompressed, all-digital audio/video interface” and “ is the industry-leading technology and de-facto standard connecting high-definition (HD) equipment, from HDTVs and personal computers to cameras, camcorders, tablets, Blu-ray players, gaming consoles, smart phones, and just about any other device capable of sending or receiving an HD signal.” ("HDMI.org FAQ"). This interface features 19 pins as shown in Figure 3 – 15. Not all of these pins are used and the HDMI connector was designed to be “future proof”. This “future proofing” allows the HDMI protocol to be upgraded without requiring a change in the hardware. With each update the protocol will make use of a previously unused pin to implement a new feature.

In an analog interface a digital signal is usually translated to a degraded and slightly inaccurate analog signal. This signal is then translated back to a digital signal to be sent to the television and displayed.

The HDMI interface tries to avoid signal degradation by transmitting a pure digital signal with no analog conversion, but some encoding. The process HDMI uses to transmit data is called Transition Minimized Differential Signaling (TDMS) (V. Wilson). TDMS will encode the signal/data to protect it from degradation while traveling through the HDMI wire. Wilson provides a brief explanation of the process:

- The transmitting device encodes the signal to “reduce the number of transitions between one (on) and zero (off)”. Each transition is a sharp drop in analog signals and while traveling, this drop can begin to wear away. This wearing will cause degradation in the signal. The encoding “protects signal quality by reducing the number of chances for the signal to degrade” (V.Wilson).
- In the physical layer of this transmission process one of the many cables in the cabling will carry the digital, encoded signal. Another cable will carry the “inverse copy of the signal”.
- The receiving device will then decode the signal. This is done by “measuring the differential between the signal and its inverse”. If there is a loss of signal along the cable then the measured differential will be used to compensate.

One of the biggest features of HDMI is its ability to protect content being transmitted from piracy. It uses an authentication protocol called High-Bandwidth Digital Copy Protection (HDCP). Each HDMI capable device contains a chip called an Extended Display Identification Data (EDID) chip to support this protocol. According to Wilson, the transmitting device, checks the authentication key of the receiving device, such as an HDTV. If both keys are accepted, the transmitting device moves on to the next step. It generates a new key and shares it with the receiving device. Ideally, this whole process, known as a handshake, takes place almost instantaneously (V.Wilson).

With the future proofing of the interface comes the ability of software and protocol upgrades/revisions. There have been four major revisions. Currently there is only two of note, revision 1.4 because it is currently being widely used by the industry and revision 2.0 which was recently released in September. With each revision come features as outlined below ("Events and Press"):
Revision 1.4

- A resolution of 3840 x 2160 at 24Hz, 25Hz, and 30Hz is now supported
- Maximum resolution is now 3840 x 2160 at 24Hz, also known as 4K television

- Ethernet over HDMI – a single data channel is dedicated to high speed bi-directional communication up to 100 Mb/sec.

These features all add up to produce key features of HDMI. These integral features include the convenience of having a single cable that transmits high definition content with accompanying audio. With high definition capability and high bandwidth, signal quality is guaranteed to be exceptional.

3.20.5 Over – The – Air Transmission

Over-The-Air Transmission, otherwise known as O.T.A. or terrestrial television, is the method of transmitting cable television over the air, technically using Radio Frequencies (RF). Over – The –Air Television was the most popular form of providing television owners with programming until the advent of Coaxial Cable Television. Cable Television Providers used to use Radio Frequencies, and still do, to provide its viewing audience with television. The way this was done, was for the viewing audience to use a radio frequency antenna, widely known as "Rabbit Ears". These antennas were nicknamed rabbit years because of the way they were able to be positioned. For one to pick up a different channel and/or try to receive a better signal, which equates to better picture quality, you would have to move the antennas.

The radio frequencies used to broadcast television programming are called Ultra High Frequency (UHF) and Very High Frequency (VHF). Ultra High Frequency is a range of frequencies from 300 Megahertz to 3 Gigahertz (WolframAlpha). Very high frequencies range from 30 Megahertz to 300 Megahertz (WolframAlpha). Ultra High Frequency and Very High Frequency Radio Waves are split into different bands for different forms of radio communications around the world. In the United States, Ultra High Frequency carries television channels two through thirteen, while Very High Frequency carries channels fourteen through eighty-three.

These antennas receive, as of 2009, digital signals over radio frequencies. The limitation of these "rabbit ears" antennas is that cable television providers and/or premium television stations, such as HBO, no longer broadcast their programming over-the-air, instead they use MPEG-2 Compressed and Encrypted digital signals over coaxial cables.

This was done in a frequency reallocation move by the Federal Communications Commission, FCC for short, in 2009. The Federal Communications Commission persuaded cable providers to switch their cable broadcasts over to a digital signal, so as to facilitate the use of new technology in high-definition television and high definition technology and the quality of programming entitled to the customers of cable providers. High-Definition television is broadcasted using the Advanced Television System Committee standard (ATSC). This standard is the standard for the broadcasting of digital television and is used in North America and other parts of the world. This move also freed up radio spectrum to be used for mobile telephone communications.

As of today, cable providers have been using the Moving Pictures Experts Group - 2 format for digital video compression, to stream said videos over coaxial cable. This was how the mandate of digital signal transmission, by the Federal Communications Commission, was satisfied. Due to the fact that Over-The-Air transmissions of cable television programming are free to be tuned to, with “rabbit ears” shown in fire 3 – 16, we will most likely aim to use one of these antennas for prototyping and testing. Due to the wide use of coaxial cable technology, antennas support the coaxial cable output and composite video output such.

3.21 Media Processor Research

This section will discuss the research done on some of the media processors we have deemed suitable for Flashback. The processor must have support with the communication mediums and the software operating environment we will choose. Everything from ARM architecture to advantages and disadvantages of processors will be discussed.

3.21.1 ARM Architecture

It was decided that, for Flashback to be responsive in respect to the Graphical User Interface(GUI) and efficient in its commercial detection algorithm, that it would need a processor that is light on power, small in size and astonishing in performance. We also need a processor that can support a 32-bit Linux Kernel based Operating System, PHY layer interfaces to I/O buses such as USB and SATA controllers. With these requirements in mind we settled on using the ARM Architecture and naturally the processors that use this architecture. The ARM Architecture, also known as A32, uses a fixed length of 32-Bits for its instruction set (ARM Processor Architecture).

The simplicity of the architecture in ARM processors have led to small-scale implementations, specifically ones that allow devices to operate with an extremely low power consumption (ARM). As cited from the ARM Reference manual “Implementation size, performance, and very low power consumption remain key attributes in the development of the ARM architecture.” (ARM) This feature is one of two reason we’re researching ARM architecture, the other reason, is that the architecture is a Reduced Instruction Set Computer (RISC), because it incorporates the typical RISC architecture features of a large “uniform register file and a load/store architecture, which means, data-processing operations only operate on register contents, and not directly on memory contents.” (ARM)

Because this architecture uses a fixed length instruction set, as mentioned earlier, the decoding of instructions is greatly simplified. The architecture also makes use of an ALU and Shifter to optimize loops. With these features of the ARM architecture, ARM processors are small in size, power consumption and program code, all while being big on performance (ARM).

Version six of this architecture, also known as ARMv6, is used in applications requiring high performance and low heat dissipation (ARM Processor Architecture). Due to these requirements being met, we decided to research any mobile processors or systems on chip using ARMv6 Architecture and higher.

From our research we found that processors using ARMv6 Architecture and higher, namely the ARM11 line of processors, have the capability of working with co-processor(s) that are included in its chip package (ARM Information Center).
An example of a co-processor is a Digital Signal Processor, which for our use-case of an ARMv6 processor or higher, is necessary for our commercial detection algorithm to be efficient. Hence we decided to use a processor that would work in conjunction with a Digital Signal Processor (DSP). What makes this co-processing setup functional and efficient is the use of shared memory and cache. Both processors will share a memory space to share data with one another in a timely manner.

3.21.2 BROADCOM BCM 2835

The BCM2835 is naturally a low-power high performance chip due to the fact that its main processor is an ARM1176JZ which uses the ARMv6 Architecture. It features a dual core VideoCore IV Multimedia co-processor which has the ability to work as a DSP to do high definition media playback, 3D graphics and high level imaging. This is perfect for our needs as one core could be used to render the GUI onto our output feed, while the other core or ARM processor could be used to support the Flashback algorithm. Since we are planning on using a SATA Hard Disk for our backing store, the fact that this chip has a built in MMU and interface for a DMA will allow the kernel easily manage our RAM and Hard Disk right upon installation of the Operating System.

3.21.3 Texas Instruments OMAP

Texas Instruments (TI) designs and manufactures a family of processors called Open Multimedia Application Platform, OMAP for short. There are currently 5 lines of subfamilies of the OMAP family, Series one through five. Currently only series three and four are available on TI’s website. OMAP processors also come pre-integrated with a number of “application-specific protocols” (Texas Instruments). The most notable features are USB with integrated PHY, an EMMC, and SATA interface.

3.21.3.1 OMAP 3

The Open Multimedia Application Platform 3 (OMAP 3) line of processors has an ARM Cortex™-A8 Multi - Processing Core and a POWERVR SGX 2D/3D graphics processor. They also come equipped with IVA-HD hardware accelerators which “enable HD, multi-standard video encoder/decoder, and 3D graphics. Some of the processors in this line even include an integrated TI C64x DSP with a max clock rate of up to 800 MHz.

These processors also come equipped with an interface that supports Low-Power Double Data Rate (LPDDR), which allows for easy installation and control of a bus for RAM modules. The chips are built using a 12x12 Ball Grid array with 45 nanometer process node (Texas Instruments).

3.21.3.2 OMAP 4

The OMAP 4 line of processors improves upon the capabilities of the OMAP 3 family of processors. Like the OMAP 3 family, the OMAP 4 family has the capability of hardware acceleration which is very beneficial when using the processor for image processor, which matches our use case. The concept of hardware acceleration is using the hardware, in the case of the OMAP family it would be one of the co-processors, to do simple, time and performance intensive instructions that would be executed a lot slower in software running on one of the co-processors. An example of such an instruction would be motion estimation which is an integral part of the encoding and decoding of MPEG-2 video files which involves the transcription of 3 dimensional vectors to a 2 dimensional plane.

The OMAP 4 family makes use of two ARM Cortex-A9 MulitProcessing Cores with clock speeds that vary in clock speed, from 800 MegaHertz (MHz) to 1.5 GigaHertz (GHz). It also has an ARM Cortex M3 for real-time instruction and I/O interrupt handling/processing which for our use case is good for fast video stream switching between our dual television tuners.

The hardware accelerator that is included in this family is optimized for video decoding and encoding which is necessary in our use case of decoding the MPEG-2 signal that we will receive from our television tuners and encoding the signal, along with our Graphical User Interface (GUI) overlay, that we will be outputting to the connected Television. The OMAP 4’s also boast a POWERVR SGX54x graphics processor which we think we could use in the future to support high definition input and output (Texas Instruments).

3.21.3.3 OMAP 5

The OMAP 5 is the latest line of media application processors from the Texas Instrument OMAP family. As stated in Texas Instrument's product description, "they are designed to support high performance applications within a low power envelope, preventing the need for fans." Naturally the features of the OMAP 5 are improvements upon the OMAP 4 line with extras such as:

- Two ARM Cortex-A15 MPCore processors capable of speeds up to 1.7 GHz each
- ARM Cortex-M4 processors for low-power offload and real-time responsiveness
- Multi-core POWERVR™ SGX544-MPx graphics accelerators
- Dedicated TI 2D BitBlt graphics accelerator
- IVA-HD hardware accelerators enable full HD, multi-standard video encoder/decoder as well as stereoscopic 3D (S3D)
- Faster, higher-quality image and video capture

3.22 User Interface Communications

This section will discuss the user's event controller and though what protocol and transmitter/receiver relationship it will communicate with Flashback. An event controller aids in the user's television experience and is essential to operate Flashback easily and universally. It should be noted that this is planned for the full extent of Flashback.

3.22.1 Infrared Remote Control

When thinking about a remote control, two types of technology come to mind: Infrared (IR) and Bluetooth. Bluetooth technology can pass through walls and solid objects because it uses radio frequency. But since "Flashback" is projected to be mainly for household uses, infrared makes the most sense. The average range of an infrared beam is about 50 feet, and the signal will not need to pass through objects to use "Flashback" in the home setting. IR usually operates between 30 kHz to 60 kHz (Brenner 1).

The remote will need certain controls, which will be transmitted through the IR LED. It will contain the following: power, selection, directional buttons, and a "prompt call" button. The "prompt call" button will call up the "Flashback" menu so that the user can decide when to activate or deactivate the process. Other button combinations will be put in order to do certain tasks, but these are the basic ones necessary for proper functions. The Figure 3 – 6 below shows the remote design our groups is considering.

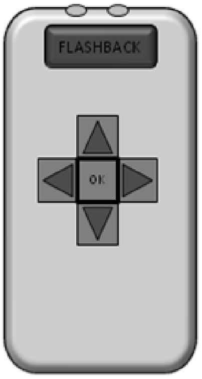


Figure 3 – 6: Remote Control Button Layout

Refer to Figure 3 – 7 below. Models being considered involve the use of the MSP430- microcontroller family or the RF4CE IC. They operate with the RC5 protocol and use ultra – low power, around “0.1 μA while waiting for user’s button-press” (Brenner 1).

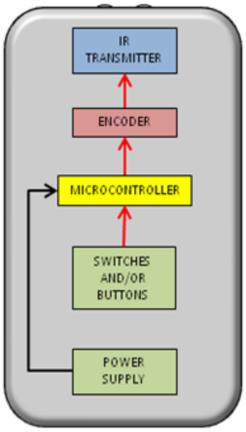


Figure 3 – 7: Remote Control General Block Diagram

3.22.2 Phillips RC-5 Protocol

Regardless of what model IR controller is being used, it will most likely use the RC-5 Protocol. The most popular has the modulation of “Manchester code (bi – phase)” and a carrier frequency of 36 kHz (Lazaridis). This frequency falls in the IR frequency range previously mentioned. There are usually 14 bits being transmitted.

The first bit is the start bit, which is always “1” and “initiates the receiving sequence (Lazaridis). The second bit is the field bit, which contains a dual purpose. We only consider that it is always “1” as well, just like the start bit, because “Flashback” most likely does not require an extended protocol, which would involve 128 commands, which translates to 7 command bits, as opposed to the normal 6 command bits. The next bit is the toggle bit, which is directly related to a button being pressed or not. The next 5 bits are command bits that will correlate to the desired device receiver, which we will decode for the “Flashback” device itself.

The last 6 bits are the command bits. The bit period, or the transmission clock is around 1778 ms, the total signal duration is about 24.892 ms, and the signal repetition internal is estimated at 100 ms (Lazaridis). Six command bits correlates to 64 commands, which is more than enough for the “Flashback” design, hence, why our design does not incorporate the 7 command bit, 128 command possibility, RC-5 extended modulation and protocol. SO we assume the field bit a “1”. Figure 3 – 19 below shows a sample message frame using the Phillips RC5 IR transmission protocol. We note the location of the MSB of the address and command bits.

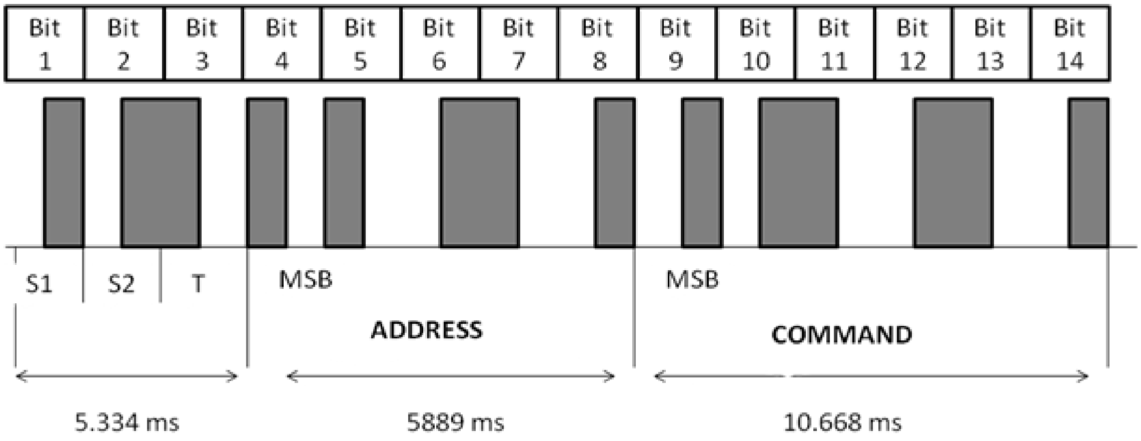


Figure 3 – 8: RC5 Example Message Frame

3.22.3 Radio Frequency Remote Control

A RF remote control would allow “Flashback” to have an “out – of – sight” controller, which in essence is not very practical, since a television would be in sight of the user, but we consider this feature because this technology involves a “low power embedded processor” (Brenner 4). Our group is considering exploring this feature. It is important to note that most RF controllers will reduce cost because of the lack of numerous discrete components.

One technology that comes to mind is Bluetooth, since internet connectivity and the radio frequency used requires a signal that will pass through solid objects. The controls on this controller will be similar to that of the in – home, infrared controller. Although this controller seems unnecessary, it is a practical way of exploring this technology and Android app configurations, and one’s cell phone can be used as an extra controller since the common remote may be misplaced.

Unlike IR, RF does not have a short transmission range and a better battery life, which usually result in a better user experience. Referring the Figure below, RF involves an additional feedback loop into the power amplifier. It helps maintain a constant output power regardless of “temperature variations” that can affect the “antenna impedance” (“How to Simplify the Design of an RF Remote Control” 3).

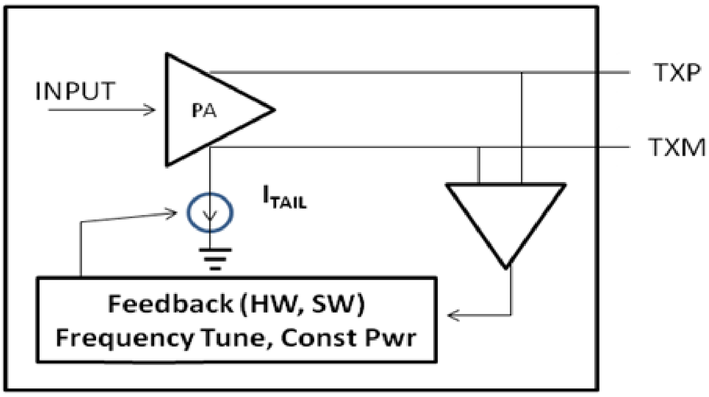


Figure 3 – 9: Additional Feedback Loop for Antenna Tuning

3.22.4 Device Controls

The “Flashback” box itself will have control buttons on them, also similar to the infrared controller. The buttons on the box will have basic controls similar to the controllers, previously mentioned. This separate control environment will aid in user experience, and will help with the user if say the remotes lose power or malfunction. Figure 3 – 10 below is just a general idea of what “Flashback” with its device controls would look like.

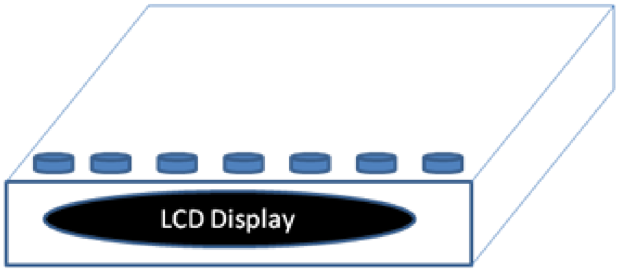


Figure 3 – 10: Flashback Device Controls

This connection may entail contact points on the printed circuit, or a separate one connected to the main layout. The buttons themselves may be a thin rubber – like sheet. Most button layouts involve a conductive disk, which would be implanted for each button on the box. The contact points would be connected to the chip it can interpret the desired command. For user interface and control confirmation, an LED could be connected to each button, and would light up to show if the contact sensor has been successfully received. The small lights would also serve as a check to whether each button is working or not, which would facilitate “Flashback” troubleshooting later discussed.

3.22.5 Seven Segment Liquid Crystal Display

“Flashback” will have an LCD display on the device itself, as previously shown in Figure 3 – 10. This LCD will be implementing the Seven Segment LCD, the properties of which are shown in the Figure below.

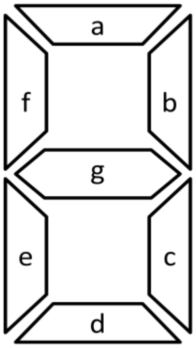


Figure 3 – 11: Seven Segment LCD Individual Segment Definition

As taught in Embedded System’s lab, this LCD logic is labeled as “old fashioned” yet effective. When a hexadecimal number is given, such as 0xFE, it would be converted to binary: 11111110. Those bits are designated for the pattern “fgedcba”, with the most significant bit being ground. “1” would represent the LED being on. “0” would be off. 0xFE would correlate to the Figure below. Also below is how the lit up LEDs can form the numbers 1 – 9. The first bit would be for “h”, not shown” which is usually a LED representing a decimal point or something along those lines, depending on the LCD design chosen.

Numbers can be moved and transformed, for example, nine could not have the “tail” (bit d), or one could be moved to the left (bits for “f” and “e” instead of “b” and “c”). Letter can also be formed, if one uses his or her imagination. Our group will, at minimal, use 4 characters, 4 – 7 segment displays, since television channels range up to the 9000s, not including the proto – channels given as “4.1 – 7” as in some cable providers. Since our group may aspire to do HD channels, we need this range. If given an opportune situation, we may include more characters to have words displayed/scrolled on or across the display such as “FLAShbACK” (note the capitalization pattern for LCD to facilitate in visual differentiation from numbers).

3.22.6 Universal Serial Bus v2.0

For the USB v2.0, there are three data rates (“Universal Serial Bus Specification” 45). They are:

- The USB high – speed signaling bit rate is 480 Mb/s.
- The USB full – speed signaling bit rate is 12 Mb/s
- A limited capability low – speed signaling mode is also defined 1.5 Mb/s

“USB 2.0 host controllers and hubs provide capabilities so that full-speed and low-speed data can be transmitted at high-speed between the host controller and the hub, but transmitted between the hub and the device at full-speed or low-speed. This capability minimizes the impact that full-speed and low-speed devices have upon the bandwidth available for high-speed devices” (“Universal Serial Bus Specification” 45). Below, Figure 3 – 25, representation of the USB cable. Since V_{BUS} is usually +5 V at the source, it will not be difficult to implement it to the power design, later discussed.

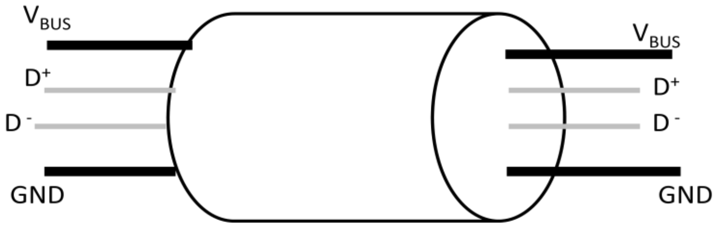


Figure 3 – 12: USB Cable

When it comes to power distribution, according to “Universal Serial Bus Specification”, the host is the one that supplies the power for use by USB devices that are directly connected, unless they are “bus – powered devices”, which are totally dependent on the power from the cable. Along the lines of power management, the host has an independent system software that “interacts… to handles system power events such as suspend or resume” (“Universal Serial Bus Specification” 46). The combination of this distribution and management allows for USB to be designed into “power sensitive systems”, like “Flashback” (“Universal Serial Bus Specification” 46).

Since the architecture of the USB is of the “plug and play” concept, hubs are a key element. Hubs are wiring concentrators and enable the multiple attachment characteristics of the USB. Attachment points are referred to as ports. Each hub converts a single attachment point into multiple attachment points. The architecture supports concatenation of multiple hubs (“Universal Serial Bus Specification” 50). A typical hub is show in Figure 3 – 26 below.

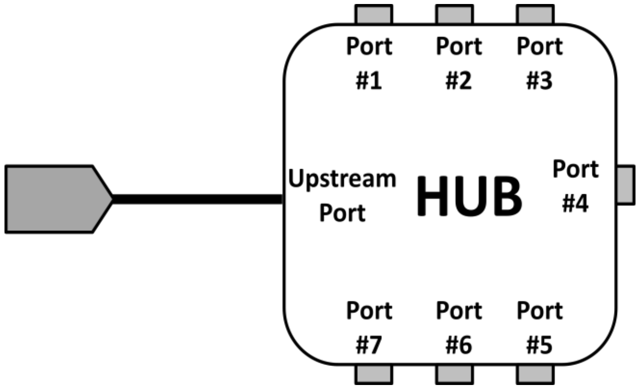


Figure 3 – 13: Typical USB Hub

“A USB 2.0 hub consists of three portions: the Hub Controller, the Hub Repeater, and the Transaction Translator. The Hub Repeater is a protocol-controlled switch between the upstream port and downstream ports. It also has hardware support for reset and suspend/resume signaling. The Host Controller provides the communication to/from the host. Hub-specific status and control commands permit the host to configure a hub and to monitor and control its ports. The Transaction Translator provides the mechanisms that support full-/low-speed devices behind the hub, while transmitting all device data between the host and the hub at high-speed” (“Universal Serial Bus Specification” 51). This type of connection and interaction may be crucial in designing “Flashback” especially when reading frames and data streams in order to differentiate commercials from scheduled programming. But from a power standpoint, this implementation is not very complex and compatible with the possible system designs. Since there is the possibility that more memory will be needed, if the RAM is not sufficient, we need the USB port in order to add external memory for functionality.

3.23 Power Electronics

This section will discuss the basic power design considerations, as well as some bock and boost devices we may implement to have Flashback operating consistently and somewhat efficiently.

3.23.1 AC-DC Converter

After using an AC-DC converter, the input voltage will be around 12V. Two viable options to implement to drop that range to the desired 5V range are a standard linear regulator, mentioned earlier, or the buck converter.

A considered buck converter circuit is the 5V output model, which can have an input voltage ranging from 8 V to 40 V. This model, using the IC LM2678, shown below and later discussed in detail, is ideal for a margin of error since the AC-DC converter being used has more versatility (“LM2678 Simple Switcher” 2).

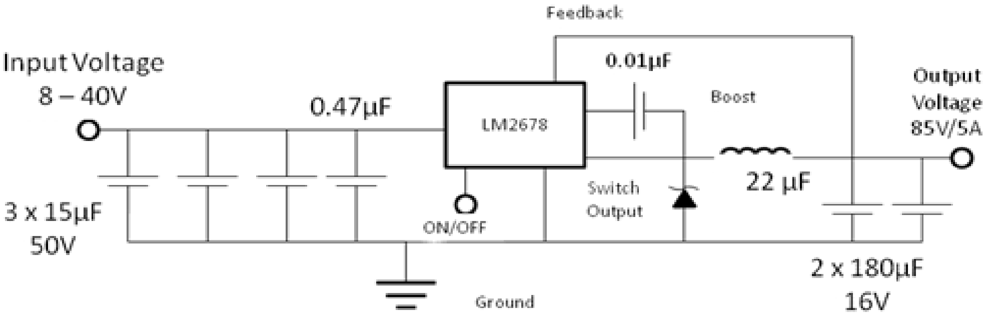


Figure 3 – 14: LM2678 Step-Down Voltage Regulator

Switching regulators are more ideal for our project because it has many advantages. Its general high efficiency is great for the power system, especially since it is compact, which aids in Flashback's final design. Since it is compact, there is no need for a heat dissipation tool, such as a fan or a copper or aluminum heat sink ("LM2678 Simple Switcher" 4). The wide input range, continuous short circuit protection, and set thermal protection are convenient when considering safety and surge protection in an emergency/unpredictable situation (power outages, sparks, emergency shutdown etc). Ideally, switching regulators typically have low ripple and noise which complements the noise of the Flyback converter, discussed in the next section.

3.23.3 Linear Regulators

There are two main types of linear regulators: standard linear regulators, and low dropout regulators, also known as LDOs. The main difference between the two is the pass element it contains and the dropout voltage, or the amount of headroom. Standard linear regulators contain a Darlington pass element, which can be of either NPN or PNP type (Day 1). Their voltage drops are as high as 2 V, which is advantageous for applications that require a large input – to – output voltage difference.

On the other hand, Low Dropout Regulators have a much lower dropout voltage, ranging to less than 100 mV. The pass element is typically an N-channel or P – channel FET, but it can also be an "NPN or PNP transistor" (Day 1). The chosen pass element drops the input voltage to the desired output voltage, which is then compared to the reference voltage by the error amplifier. It then drives the voltage to the gate, connected to the FET's gate, which is the same node as the output voltage.

Figure 3 – 15 below is the block diagram for the LDO power reference. They require a low minimum operating voltage, can have high efficiency, and low heat dissipation if used correctly. The thermal dissipation for LDOs can be very high if used incorrectly since input voltage will be fed at a constant rate and build unless there is some sort of protection. The efficiency is essentially voltage out over voltage in so if there is a large differential then the LDO will raise the temperature of the Flashback device significantly.

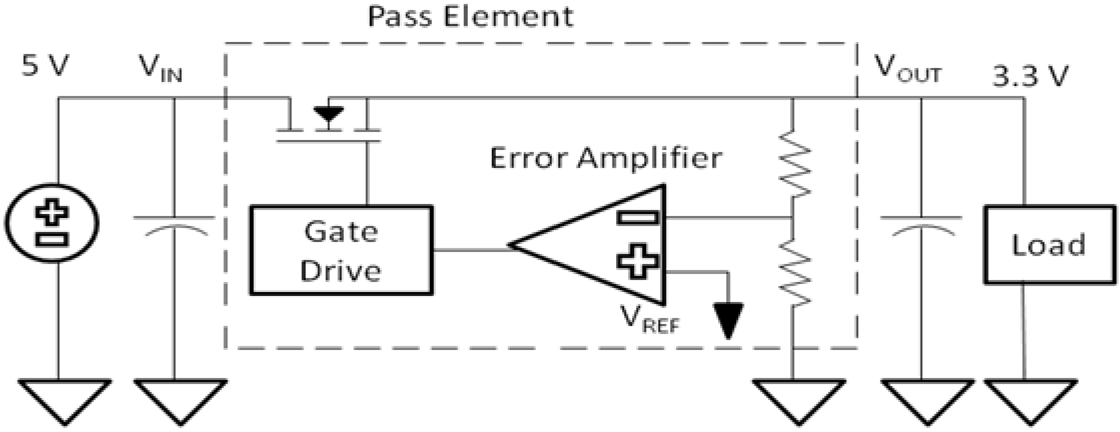


Figure 3 – 15: LDO Block Diagram

For “Flashback”, the power system will require both standard and LDO regulators. Since the system we are making will need power ranging from 1.8 V to 5.5 V, we will use an AC-DC converter, most likely 120V AC to 12V DC and use a standard linear regulator combination to bring it with 7 V, or around that range, and use a variety of linear dropout regulators in a parallel combination to feed power to each element of the box.

LDOs consist of a power FET and a differential amplifier that utilizes open-collector, open-drain rather than collector and emitter. They can also be used as filters. The way our Flashback device will utilize LDOs is by first implementing a buck converter to step-down the voltage and then use a LDO to clean up the new step-down voltage and reach the desired input voltages of 5 V, 3.3 V, and 1.8 V, previously mentioned.

3.23.4 Non – Inverting Operational Amplifiers

Non – inverting operational amplifiers, like the ones discussed in Electronics II lab, can be assessed to this power design. It would be advantageous because they will create a sense of familiarity among Flashback’s power design.

The general sense of this alternative design would implement another step down voltage regulator after the AC – DC converter and then connect an operational amplifier combination to that small voltage to reach the desired voltage. Since a non – inverting amplifier has a simple closed – loop voltage gain ratio of (1+ $R_2/R_{1)}$; it will be a straightforward circuit to design and analyze (Neaman 639). Refer to the Figure below for the general non – inverting operational amplifier.

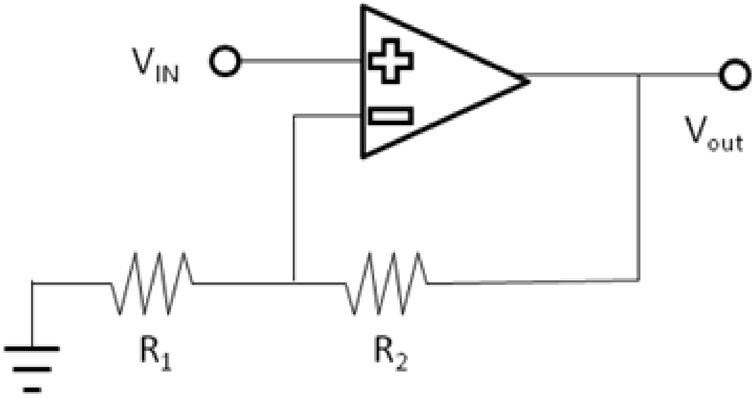


Figure 3 – 16: Non – Inverting Operational Amplifier

Although this other design would be non – complex and thrifty, it lacks efficiency. Therefore, our group deemed this an “emergency design” for Flashback if the other designs do not work, since one of the goals of this project is to tamper with other technologies. Plus the point of an amplifier is to amplify voltage, no decrease it; a simple resistor combination can do this. Our group is very familiar with the TL081 – TL084 family of operational amplifiers. This design may be implemented in one of the parallel power rails just to incorporate it and display proper execution.

3.23.5 Flyback Converter

The Flyback converter is usually of the buck boost principle, which amplifies DC voltages from say 12 V to 180 V. From an efficiency standpoint, it makes no sense for “Flashback” to even consider, let alone incorporate this into the design, seeing as the aforementioned design is basically an AC – DC conversion from 120 DC V to 12 AC V. Then our group would incorporate a buck converter to lower the 12 V to the projected desired voltages of 5 V, 3.3 V, and 1.8 V (or voltages of that range). There is, however, a more complex flyback IC where the output voltages will be within the desired range and quite a bit more efficient and less costly. This alternative design is under consideration for the “Flashback” device.

This design, later discussed in detail, involves an input of a DC voltage of 120 V, which correlates with a standard wall outlet, and an dual output of +5 V and +12 V. Reconfiguring the circuit design, our group can achieve dual output voltages closer to our desired voltages, or simply use a buck converter, mentioned in the section above, to get to the voltage range “Flashback” requires. The new combination design being considered may be more cost and power efficient, but will take up more space, which may be a problem seeing how the design is starting to pan out.

3.23.6 FET Load Switch

With loads powered by voltages ranging from 1.8 V to 5 V, a load switch with integrated hot – swap management and protection would be ideal. This device would facilitate power management by preventing load shorts from interrupting or damaging other system components. "Thermal shut down protection that prevents damage to the device when a continuous over – current condition causes excessive heating by turning off the switch" would also help with heat dissipation, later discussed, and direct interfacing ("Ultra – Low Power, Low Input Voltage, Current – Limited Load Switch with Shut – Off, Auto – Restart, and Over – Current Condition Time – Out" 1).

The load switch can also be used to protect the LDO, previously discussed. "The inrush current required by the LDO at startup…can exceed the current limit and initiate a blanking condition" ("Ultra – Low Power, Low Input Voltage, Current – Limited Load Switch with Shut – Off, Auto – Restart, and Over – Current Condition Time – Out" 1). Having a load switch would help placate this scenario by temporarily increasing the current limit until it regulates and normalizes itself. Adding this small, low current component will improve the efficiency and performance of the power system as a whole.

3.23.7 Electrostatic Discharge Protection

Electrostatic discharge, or ESD, protection is a safety measurement that "prevents possible damage to the hardware and decreases any possible impact to your operations" ("Understanding ESD requirements"). ESD is a sudden flow of electricity caused by "contact, an electrical short or dielectric breakdown" (ESD Protection Datasheet). Lighting is an example of ESD, but at a much larger scale. ESD – sensitive parts in "Flashback" require special attention.

Some precautions must be taken with these ESD – sensitive parts. These parts should be set aside, and after putting on the ESD straps, our group must refrain from "discharging paths by…not allowing the ESD – sensitive parts on the machine cover and avoid contact with large metal objects" ("Understanding ESD requirements). It is also pertinent that one minimizes the static electricity present. When ordering and configuring parts for "Flashback", our group must be aware of specialized plots regarding this manner in order to select the right protection element for our desired system – level design.

3.24 Audio Line Driver

Transformers are a traditional input or output audio line – coupling element. Transformers are vulnerable to picking up noise, they have a poor frequency response, pick up distortion, and have a limited operating level. An audio line driver is a solution to transformers problems. An audio line driver is an amplifier used to improve the strength of a signal at its source by driving the input to the transmission line with a higher than normal signal level (Jung and Garcia). Since the signal is amplified, there will be less signal loss as it is transmitted over longer sections of a cable (Tatum and Harris). Audio line drivers reject common-mode noise that may typically be picked up on a twisted pair transmission line. This will increase the quality of the signal over a long cable. In the case of mobile audio, the use of a line driver allows an amplifiers gain to be set lower which will in turn reduce low-level noise (Jung and Garcia).

3.25 Related Products

This section will discuss related products out there that may suffice as reference design and algorithm coding. We do not intend on stealing or borrowing the designs, but instead learn from them to create our Flashback.

3.25.1 Comskip

Comskip is a software application used for detecting commercials within MPEG files. The application was created for the windows platform and officially isn't housed in a device. The triggers that Comskip uses to determine if a commercial break has occurred are black frames, silence, and change in aspect ratio, which are all applicable to Flashback as well.

Comskip does not contain the functionality to skip commercials since its implementation was created for one input of streaming video. The software's main purpose is to analyze an MPEG video file and pin-point the time in which these commercial breaks occur. After Comskip has completed its algorithm, the software produces an output file for the user to access and read.

This application was intended to be a plugin for GB-PVR. GB-PVR is a media centre created for the use on the windows platform that allows for the sharing of media files on devices that are on the same network. When Comskip is implemented into the GB-PVR users can view information about the frames they are viewing by accessing the tools that Comskip provides. The instruments that Comskip contains can let the user know what frame number they are currently on, what the aspect ratio of the frame is, and information on the television station's logo that the file was recorded from. Comskip also incorporates a graph feature that portrays all of these different attributes of a frame.

The Comskip interface also contains a timeline widget. This timeline widget contains multiple levels that represent the data of a frame. The information represented in these notches are the levels of volume in that frame which appear as green pixels, the time frame in which black frames were detected which appear as red pixels, white pixels will be displayed when a logo is absent and black pixels when the logo is visible. The color coding of pixels in this widget make for an easy recognition of these different commercial break triggers.

This related product is a great structure to model our Flashback devices software from. It contains all of the triggers we're determined to add to our algorithm and even presents the algorithms findings in a colorful timeline. The only thing Comskip doesn't do is analyze life streaming television which is the main purpose for our product.

3.25.2 ReplayTV

ReplayTV is a digital video recorder that fashions many user friendly functions. Some of the specifications and functions that ReplayTV contains are; larger buffer to store a recording of the current channel that is live, various functions thats allow the user to replay a recently viewed scene with the touch of a button, navigate through a program quickly and effectively, and skip through certain portions of a program that aren't important to the user. ReplayTV also sports the ability to stream television into any television in the house, allowing users to start a program in one room and finish it in another without the hassle of fast forwarding through a recording. The device scans the bit stream from the coaxial cable at a rate of 480P for HDTV. The speed of scanning on this device is the reason why this device is advertised as the most powerful DVR.

ReplayTV tokens the saying of “being the first DVR to put control back into the customer's hands” ("Why ReplayTV"). Flashback identifies with this saying since one of the goals we wish to accomplish with its creation is allowing users to take back the control of their living rooms. ReplayTV succeeds in returning control to their users by the implementation of QuickSkip, Show|Nav, Instant Replay, and Jump Anywhere.

Quickskip's intention was to serve as an easy way for users to skip over a program 30 seconds at a time. If a user wishes to fast forward through a program on normal DVRs, the user will have to select fast forward and change the frame rate until they reach a section that's of interest to them. Commercials tend to be about 30 to 45 seconds each making Quickskip a great tool for skipping over commercials and easily coming to a stop.

Show|Nav's purpose is to serve as a way for users to easily browse through shows in pursuit of finding the portion of programming that the user is looking for. This function divides a program into sections so the user can easily skip through multiple sections. The functionality is similar to how movies are divided into sections on DVD. Sifting through sections of programming allows users to view what they want in a timely manner.

Instant Replay's functionality involves replaying television instantly. The requirements for the program do not require for the viewing to be recorded, it can be live as well. The purpose that this function serves is for users to quickly rewind the most recently viewed frames in case the user got distracted or didn't comprehend what actually occurs.

Jump Anywhere is meant to be used in succession to QuickSkip. This function is a way to jump to any specific portion of a television show or movie. This functionality is accessed by pressing a number on the remote control after an instance of QuickSkip.

The power and functionality of ReplayTV is what we intend to match in Flashback. The memory buffer size, response time, and over all control given to the user are all considerations to be researched for Flashback.

3.25.3 Autohop

Autohop is Dish Network's answer to a commercial skipping product. This product has the functionality of a set top box and a digital video recorder. It can stream television from a cable source or play shows and movies on applications through the built in Wi-Fi of the device. This device also interfaces with smart phones, tablets and computers.

Sling is the component in the Hopper that gives the product internet connection. With this component in active, users can stream their recordings from the internet using either an Ethernet cable or the Wi-Fi. This component allows the Hopper to record up to six shows simultaneously. Sling gives the Hopper the feel of a smart device due to its ability to not only stream video from the provider or from the internet but also allows users to socialize through their hopper with twitter or view any recording on the hopper from any device on the network.

The requirements of the program that a user may wish to try to hop through commercials are; programs that are recognized as "Primetime Anytime", and programs that are recorded. "Primetime Anytime" programs are programs that have been aired on ABC, CBS, NBC, and Fox. Any program from these four stations must be downloaded enable commercial hopping. This functionality is similar to what we're trying to achieve with Flashback in the aspect of commercial skipping on recordings.

Although the Autohop contains many other features that are out of the scope of the specifications currently set for Flashback, the main feature we're interested in is the commercial hopping. On the Autohop when a user's program fits the criteria for a show in which commercials can be skipped, the user is prompted to select whether or not they wish to skip commercial breaks. This similar idea is what will be implemented in our Flashback device; it's all about giving the user the choice to decide how they wish to experience their shows.

4.0 Project Hardware and Software Details

The section will discuss all hardware and software decision made from the research done in precious sections. Note that all sources, environments and hardware are compatible and support one another. Hardware processors must have the potential to handle the desired IC's and components.

4.1 Video Signal Input / Source

To prove the concept of Flashback, our group will focus on Over the Air TV transmission since this medium will provide readily available television signals for our use. In the future, Flashback will pair with cable sponsors that encrypt their television signals. With access to their decryption algorithms, we will be able to implement Flashback features with their device.

4.1.1 Over – The – Air Television

To receive Over the Air programming, our group decided to use an antenna that pairs with Hauppauge WinTV 950Q tuner and supports reception of Ultra High Frequencies and Very High Frequencies. The antenna is amplified so as to give us the capability of receiving as many channels as geographically possible. Compared to non-amplified antennas, an amplified antenna gave us a larger sample set for testing of Flashback due to the amplified antenna's ability for receiving more channels. This antenna uses RG-6 coaxial cable and the F connector as its interface with the TV tuner.

Our group has selected the TV antenna that comes with the Hauppauge tuners, with the intention of one day upgrading to an outdoor antenna for larger testing. With this antenna our group was able to receive 54 channels. Table 4 – 1 depicts some channels we were able to receive from our antenna in the Orlando, Florida area.

Call Sign	RF Channel #	Virtual Channel #	Network Name
WESH-DT	11	2.1	NBC
WKCF-DT	17	18.1	CW
WOFL	22	35.1	FOX
WUCF-TV	23	24.1	N/A
WKMG-DT	26	6.1	CBS
WRDQ	27	N/A	IND
WFTV	39	N/A	ABC
WOTF-DT	43	43.1	Telemundo
WOPX-TV	48	56.1	ION

Table 4 – 1: OTA Channels in Orlando. Information provided "TV Fool"

4.2 Hauppauge WinTV-HVR-950Q Tuners

As discussed earlier, Flashback required a dual television tuner that is able to handle both analog and digital incoming signals. Our group first looked into the use of TV tuner IC's. One of the IC's we found was a part from Silicon Labs. After I tried to place an order for a mere handful, the supplier told me that they reserve these parts for top tier production companies and we couldn't even obtain a sample. Then we moved onto a different IC from CrestaTech. I spoke with a CrestaTech engineer about incorporating their IC into our project and he informed me that we wouldn't be able to successfully implement the device unless we also built a digital demodulator. He suggested we take a look at coax to USB TV tuners. Our group settled on the Hauppauge WinTV-HVR-950Q USB TV tuner.

The Hauppauge tuner can support NTSC, ATSC, and Clear QAM digital cable TV standards. This device incorporates both the TV tuner ICs that our team was looking for and the necessary demodulators. As a bonus, our software team members were able to locate software support to produce a video stream for Over the Air transmission in Orlando, FL. By incorporating two of these devices, our project was able to capture the necessary two television channels needed for Flashback concept. ("Watch, Pause and Record TV on Your Windows PC or Laptop.").

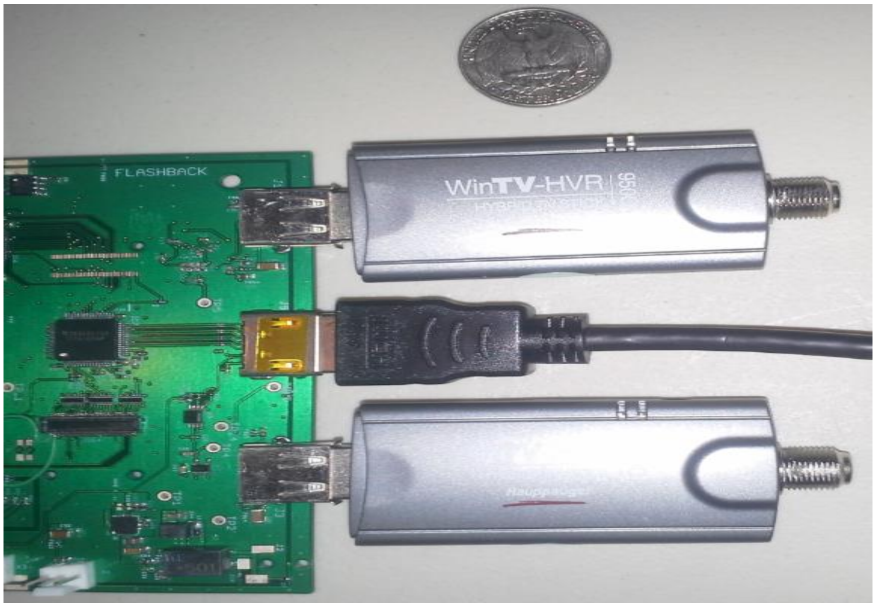


Figure 4 – 1: TV Tuner Connection

4.3 TV Tuner Input

The team used two USB-A 2.0 ports to plug these tuners into our board as shown in Figure 4-1. To control the common mode data line input noise, choke filters were placed on both sets of data lines as seen in schematic figure x-xx. After our board accepted the two TV tuner inputs, we needed a way to control the two different sets of data lines. In order to accomplish this feat, we added a USB 2.0 High-Speed 4-Port Hub Controller from SMSC. Even though this device is built for up to four ports, we will only be using two of the four possible ports. In addition, the USB hub controller has fully integrated USB termination and Pull-up/Pull-down resistors, enhanced EMI rejection, and ESD protection. The availability of these features already incorporated in the chip reduces the board space required for USB control, in turn, lowering our overall cost. A USB hub essentially allows the host to send information to any and all of the 4 ports but if one of the downstream ports was to send information back to the host then only the host would see the data. Since we used this to control the TV tuners inputs, they are both able to send data to the processor without sending them to the other TV tuner ("USB 2.0 Hi-Speed Hub Controller "). The schematic figure x-xx shows how we implemented the USB Hub Controller after the input from the ports.

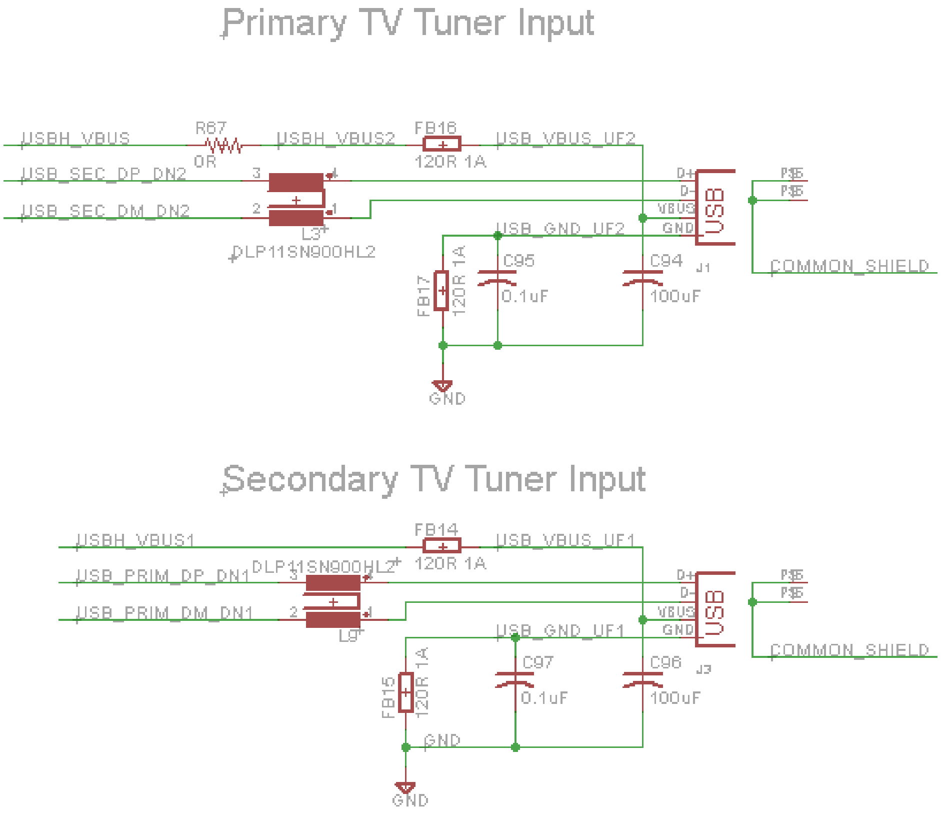


Figure 4 – 2: TV Tuner Schematic

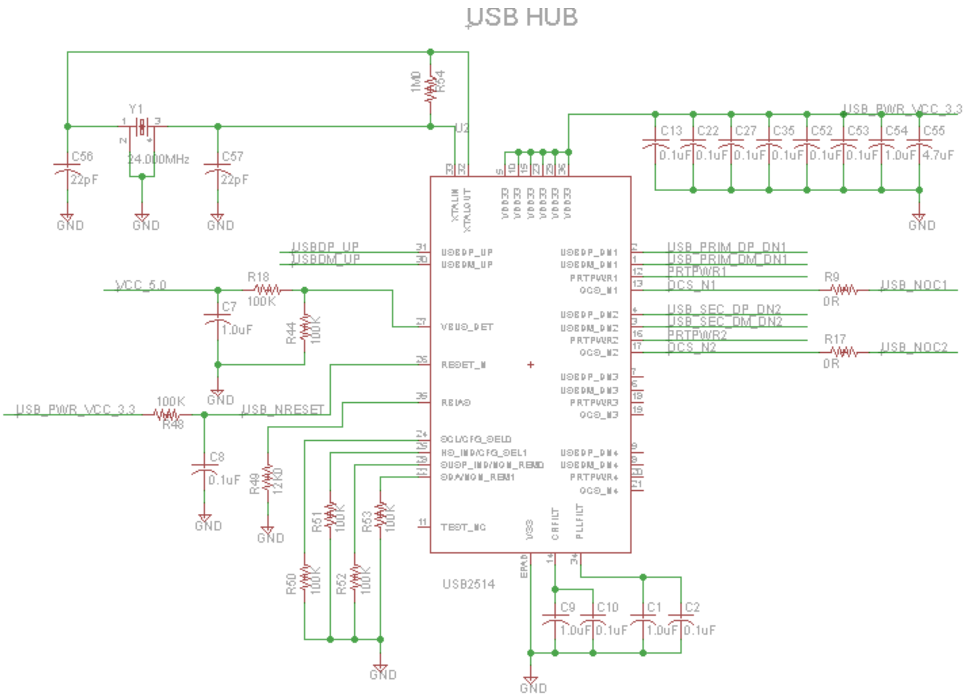


Figure 4 – 3: USB Hub Schematic

4.4 Gumstix Overo® TidalSTORM

Our team used the Gumstix Overo® TidalSTORM computer on module, hereinafter refered to as the TidalSTORM. The TidalSTORM uses a Texas Instruments DM3730 processor with ARM Cortex-A8 architecture operating up to 1-GHz. The DM3730 is also compatible with Texas Instruments OMAP™ 3 Architecture. This processor was designed for Digital Media applications. The processor contains a high performance Image, Video, Audio (IVA2.2™) subsystem and a POWERVR SGX™ graphics accelerator ("DM3730, DM3725 Digital Media Processors").

IVA2.2 subsystem functions up to 800-MHz DSP core but can operate on multiple frequencies. In addition, the subsystem has an Enhanced Direct Memory Access (EDMA) Controller with 128 independent channels and video hardware accelerators ("DM3730, DM3725 Digital Media Processors"). Normally the CPU would read byte by byte of a data and then write each byte to memory. An EDMA controller allows a bypass of the CPU to transfer data quickly and directly from the I/O to memory (Rai).

POWERVR SGX™ graphics accelerator has a tile based architecture delivering up to 20 MPoly/sec ("DM3730, DM3725 Digital Media Processors"). Tile based architecture uses rendering techniques that eliminate redundancies found in the

3D pipeline. Therefore, tile based architectures significantly reduce memory bandwidth limitations that restrict rendering. Tile architecture renders first groups polygons together in groups called display lists. These display lists allow a scene to be broken into smaller blocks, known as tiles, which are rendered independently ("STMicroelectronics Kyro II 64MB."). The graphics accelerator also has a universal scalable shader engine with multi-threaded engine incorporating pixel and vertex shader functionality, API support for OpenGLES 1.1 and 2.0, fine grained task switching, load balancing, power management, and programmable high quality image anti-aliasing (“DM3730, DM3725 Digital Media Processors”).

TidalSTORM interfaces with our expansion board over two 70 pin connectors summing to 140 total pins as shown in schematic figure x-xx. These 140 pins provided the necessary functionality for our device. We were able to incorporate DVI-D, I²C (i.e. EEPROM, HDMI, and LCD), UART for Infra-Red and USB-B, audio output, standard definition video, USB input from HUB, and Ethernet.

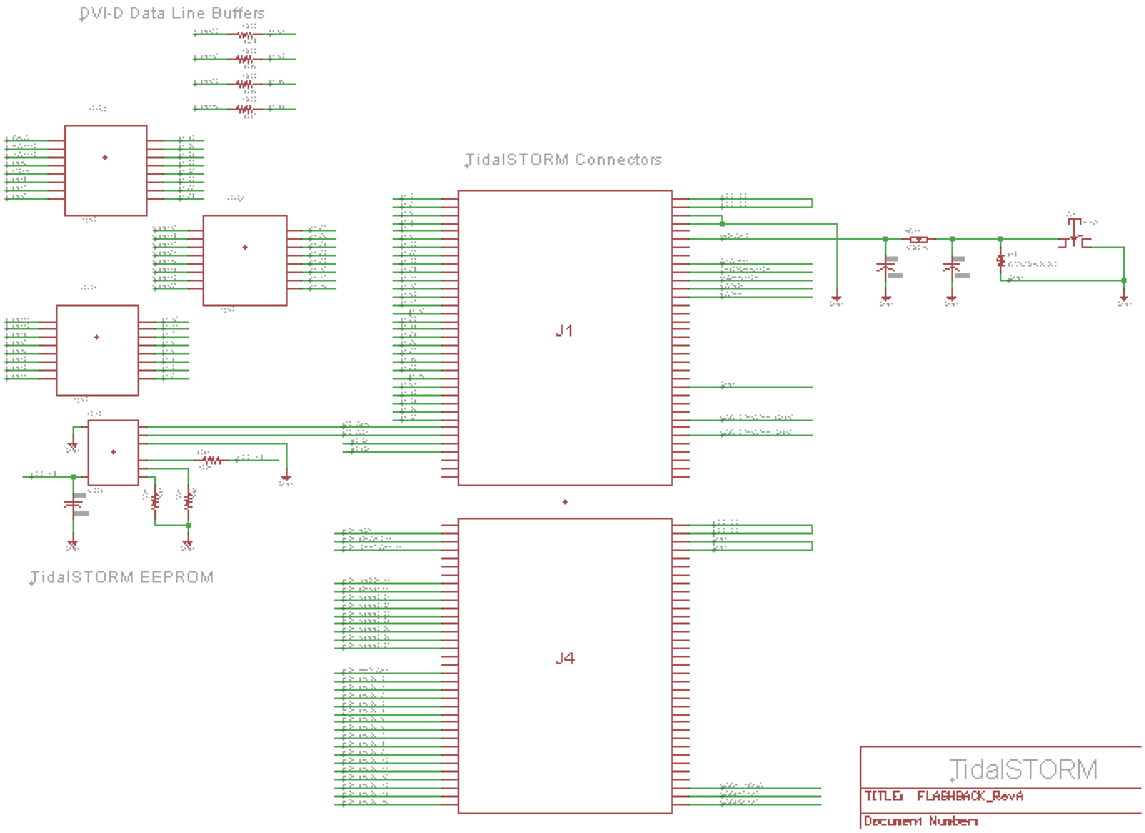


Figure 4 – 4: TidalStorm Connector Schematic

4.5 MicroSD Card

Our group has decided that to meet our speed specifications that we will use a solid state drive for storage. This drive will store our selected operating system, user data, and recorded television programming. For a solid state drive to be considered by our group, we have decided that they need to meet these minimum specifications:

- 32 Gigabytes of available storage
- Serial ATA II interface
- No more than $80 in price
- Must be new
- Multi-Layer Cell Architecture
- No more than 3 Watts of power consumption
- Internal drive
- No more than 2.5'' in size

With that in mind we are considering the following solid state drives:

ADATA Premier Pro SP600	
Model #	ASP600S3-32GM-C
Form Factor	2.5"
Capacity	32 GB
Interface	SATA 3
Flash Memory	Multi-Layer Cells
Power Consumption (Active)	1.5 Watts
Drive Type	Internal
Sequential Read	360 MB per second
Sequential Write	130 MB per second
Average Price (USD)	$45.00

Table 4 – 2: Specifications of ADATA SSD. Information retrieved from Newgg.com

Crucial v4	
Model #	CT032V4SSD2
Form Factor	2.5"
Capacity	32 GB
Interface	SATA 2
Flash Memory	Multi-Layer Cells
Power Consumption (Active)	N/A
Drive Type	Internal
Sequential Read	200 MB per second
Sequential Write	60 MB per second
Average Price (USD)	$40.00

Table 4 – 3: Specifications of Crucial v4 SSD. Information from crucial.com

Kingston SSDnow	
Model #	SV300S37A/60G
Form Factor	2.5"
Capacity	60 GB
Interface	SATA 3
Flash Memory	Multi-Layer Cells
Power Consumption (Active)	2.052 Watts
Drive Type	Internal
Sequential Read	450 MB per second
Sequential Write	450 MB per second
Average Price (USD)	$68.00

Table 4 – 4: Specifications of Kingston SSDnow. Information retrieved from kingston.com

SanDisk Ultra Plus 2.5"	
Model #	SDSSDHP-064G-G25
Form Factor	2.5"
Capacity	64 GB
Interface	SATA 3
Flash Memory	Multi-Layer Cells
Power Consumption (Active)	2.052 Watts
Drive Type	Internal
Sequential Read	520 MB per second
Sequential Write	155 MB per second
Average Price (USD)	$68.00

Table 4 – 5: Specifications of SanDisk Ultra Plus 2.5". Information retrieved from sandisk.com

Our group has decided that when we are ready to purchase parts for prototyping that we will most likely purchase the Crucial CT032V4SSD2 for testing. Then our group will switch to either the SanDisk SDSSDHP-064G-G25 or Kingston SV300S37A/60G for a final marketable product since these drives have a larger capacity.

4.6 User Interface

This section will discuss the intended event controller and the protocol it will implement. We consider the necessities and realistic situations of the controllers use to better infer what is needed to avoid excessive cost and complexity. Ideally, a common remote application would be best suited

4.6.1 Infrared Sensor

An infrared sensor will be implemented to read the user's input from the event controller's IR transmitter. The IR sensor Flashback imposes is the TFBS4711, shown in the Figure 4 – 5 below. Its small package size of 1.9 x 3 x 6 mm, transmit distance of 8 m, tri-state receiver output, and operation range from 2.4 V to 5.5 V over a wide temperature range from -25 degrees Celsius to 85 degrees Celsius makes this model ideal for Flashback's operation, as well as being a simple, flexible component to add to the printed circuit board.

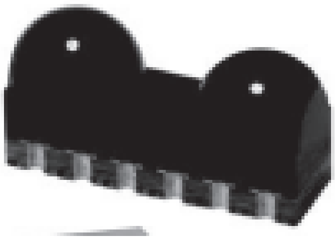


Figure 4 – 5: TFBS4711

Table 4 – 6 below shows the pin description of the TFBS4711.

Pin #	Symbol	Description	I/O	Active
1	V_{CC2} IRED anode	Connect IRED anode directly to the power supply. IRED current can be decreased by adding a resistor in series between the power supply and IRED anode		
2	TXD	This Schimiti-Trigger input is used to transmit serial data when SD is low. AN on-chip protection circuit disables the LED driver if the TXD pin is asserted for longer than 110 µs	I	HIGH
3	RXD	Received data output, push-pull CMOS driver output capable of driving standard CMOS or TTL loads. No pull up or pull down resistor needed	O	LOW
4	SD	Shutdown. The input threshold voltage adapts to and follows the logic voltage swing	I	HIGH
5	V_{CC1}	Supply voltage		
6	GND	Ground		

Table 4 – 6: TFBS4711 Pin Description

When it comes to board design and layout, not many external components are needed. The recommended circuit diagram is shown Figure x – x, below. As noted in the TFBS4711 datasheet, capacitor C1, 4.7 µF16 V, is buffering the supply voltage while eliminating the inductance of the power supply line. Resistor R1 is the current limiting resistor, whose recommended value depends on the current to be adjusted. Capacitor C2 paired with resistor R2 is the low pass filter for smoothing the supply voltage

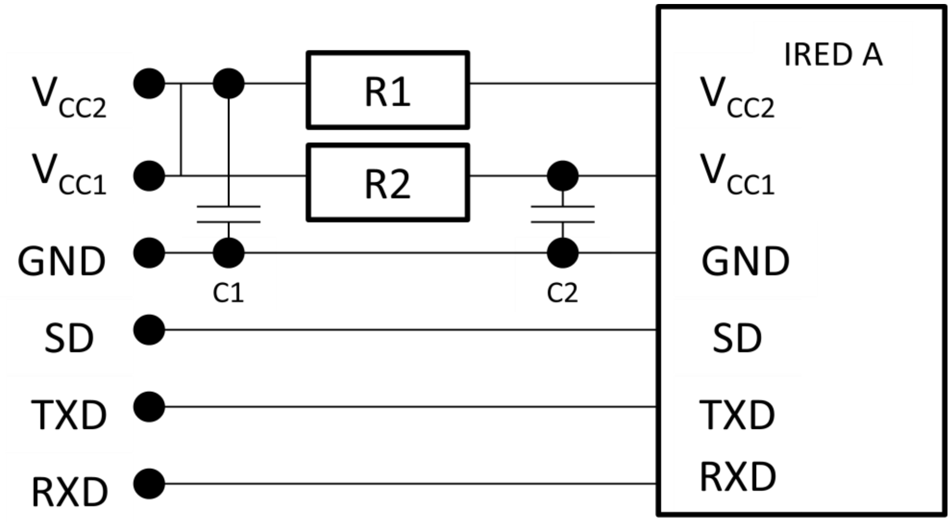


Figure 4 – 6: Recommended Circuit Diagram

4.6.2 Liquid Crystal Display

Flashback uses Adafruit's 1.2" 7-segment backpack. This LCD was chosen because not only did it operate on either 3.3 volts or 5 volts, as board uses, but also because it 5 inch characters are easily seen for user confirmation of event controller command. Figure x – x below shows the backpack. Instead of using individual pins for each LED, this communicates over I2C and comes with firmware, so that implementation and coding can be done very easily.

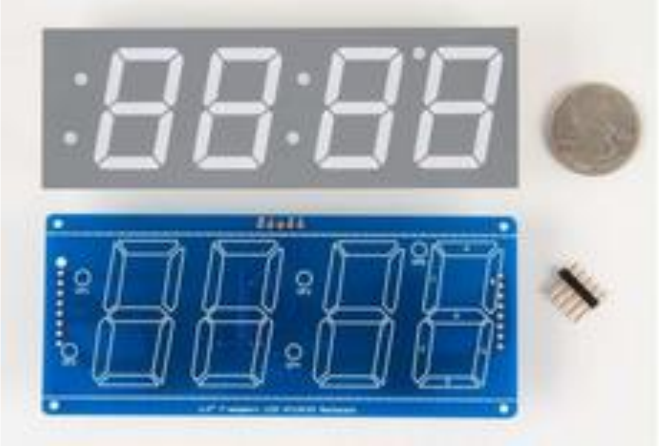


Figure 4 – 7: Four Character 7 Segment Display

We note that this implementation is already assembled and just requires a 5 pin header. The 5 pins are Pin D: I2C Data Pin (SDA), Pin C: I2C Clock Pin (SCL), Pin +: 5V, Pin -: Ground, and Pin IO: I2C Bus Voltage. The flexibility to use this on a 3.3V and 5V processors also gives us flexibility in future designs. Since the

DM3730 is a 3.3 V processor, the pins are laid out as follows: D = SDA, C = SCL, + = 5 v, - = GND, and IO = 3.3 V.

4.7 DVI-D

Flashback device supports a Digital Visual Interface-Digital instead of HDMI because of pin availability over the TidalSTORM connectors. The difference between DVI-D and HDMI are audio signal lines. We implemented audio through audio lines that are connected to two external speakers since these audio lines don't meet the HDMI specifications. Our device is still able to provide 24-bit true color through the use of TFP410, a Texas Instruments *PanelBus*™ Digital Transmitter. This IC has a few key features to improve noise performance, such as, enhanced PLL noise immunity, enhanced jitter performance with no HSYNC jitter anomaly and a negligible data-dependent jitter. The TFP410 also contains an I^{2}C host interface to allow enhanced configurations in addition to power-on default settings programmed by pin-strapping resistors. To send data out of the IC and to an HDMI connector, we had to add a level shifter from 1.8V to 5V to satisfy HDMI standards. In addition, we added common-mode choke filters and ESD protection circuits on data lines going to the port. The addition of these components provided a clean output signal on the HDMI port. ("TFP410"). Please see the schematic figure x-xx for how Flashback device outputs DVI-D.

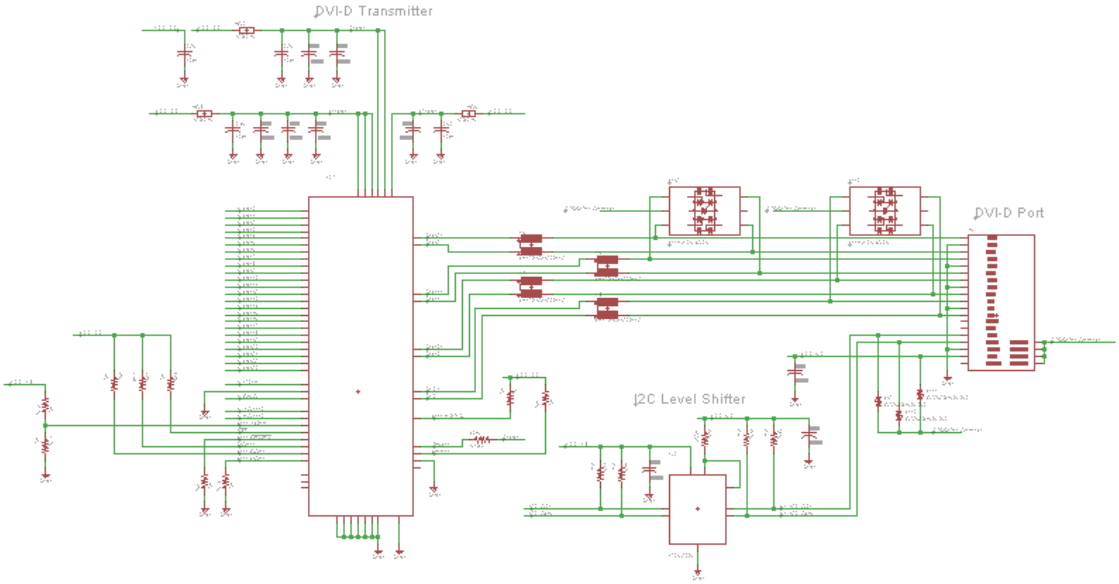


Figure 4 – 8: DVI-D Schematic

4.8 Standard Definition Video Filter

The TidalSTORM outputs a chrominance and a luminance signal over one of it's 70 pin connectors. First, a filter was added to improve both the signal strength and the quality. The part we chose for this operation came from Texas Instruments and it is their THS7315 3-Channel SDTV Video Amplifier with 5th-Order Filters and 5.2-V/V Gain. This part was recommended by TI to operate alongside their digital media processors. Although this device has three channels, we only needed two (chrominance and luminance). This device uses a bandwidth catered to televisions 6 MHz bandwidth requirement and provides a 14.3 dB gain for any potential loss in the hardware path. This part runs on our 3.3V power plane and has a max quiescent current of 15.6mA ("3-Channel SDTV Video Amplifier with 5th-Order Filters and 5.2-V/V Gain ").

A composite video signal, standard definition video, is the combination of chrominance and luminance signals into their respective bandwidth. To combine these signals a capacitor of 470pF was added to shift the chrominance signal before it was added to the luminance signal to create a single composite signal. The entire standard definition video signal filter is shown below in the schematic Figure 4-9.

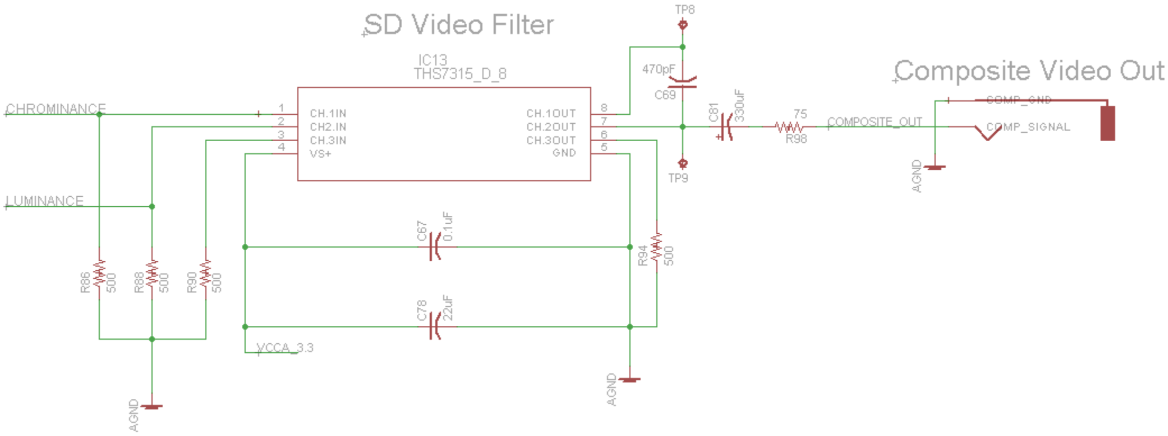


Figure 4 – 9: SD Video Filter Schematic

4.9 Audio Line Driver

After comparing multiple audio line drivers, the DRV601 2-Vrms DirectPath™ pop-free variable input gain line driver designed by Texas Instruments is the best fit for our application. DirectPath™ operates from a single supply but makes use of an internal charge pump to provide a negative voltage rail. This design is capable of driving 2 Vrms into a 600-Ω load at 3.3 V power supply. In addition, this package is designed to allow the removal of output DC-blocking capacitors. The removal of DC-blocking capacitors helped minimize board space, eliminate degradation of low-frequency response, and save costs on components and printed circuit board. In addition to a reduced component cost the audio line driver uses thin quad-flat no-leads (QFN) package with 4 mm x 4 mm dimensions and 20 pins. The small package size allowed the group to design a smaller sized PCB and save cost. Through the use of a resistive network, the device can also support a variable gain range and line outputs that are ESD protected. Furthermore, this device has the ability to independently shut-down the right and left audio channels. Lastly, Texas Instruments offers free samples for this product.

The recommended operating conditions of the driver will require three power rails. The supply voltage recommended range is from 1.8 V to 4.5 V. The high-level input voltage specifies a minimum of 1.5 V and the low-level input voltage specifies a maximum of 0.5 V. The supply current will depend on the operating voltage but if the device is being powered at max voltage then the supply current will be 8.7 mA. Therefore the total power dissipation will be 39.15 mW. The DRV601 DirectPath™ line driver amplifier required power supply decoupling to ensure that the noise and total harmonic distortion (THD) are low. With Texas Instruments suggestion, we placed a good low equivalent-series-resistance (ESR) ceramic capacitor, 22 µF, as close as possible to the device V_DD lead. Here are some important notes we found in datasheet:

- The signal and power ground must be routed to the decoupling capacitor separately in order to provide proper device operation.
- The gain setting resistors must be placed close to pin 13 and 17 to minimize capacitive loading effects.
- The exposed metal pad must be soldered onto the PCB but not connected to ground or power planes.

All of the information found in this section was provided through Texas Instruments DRV601 datasheet ("DIRECTPATH™ STEREO LINE DRIVER, ADJUSTABLE GAIN"). Please reference the schematic figure x-xx for how we implemented the information in above text.

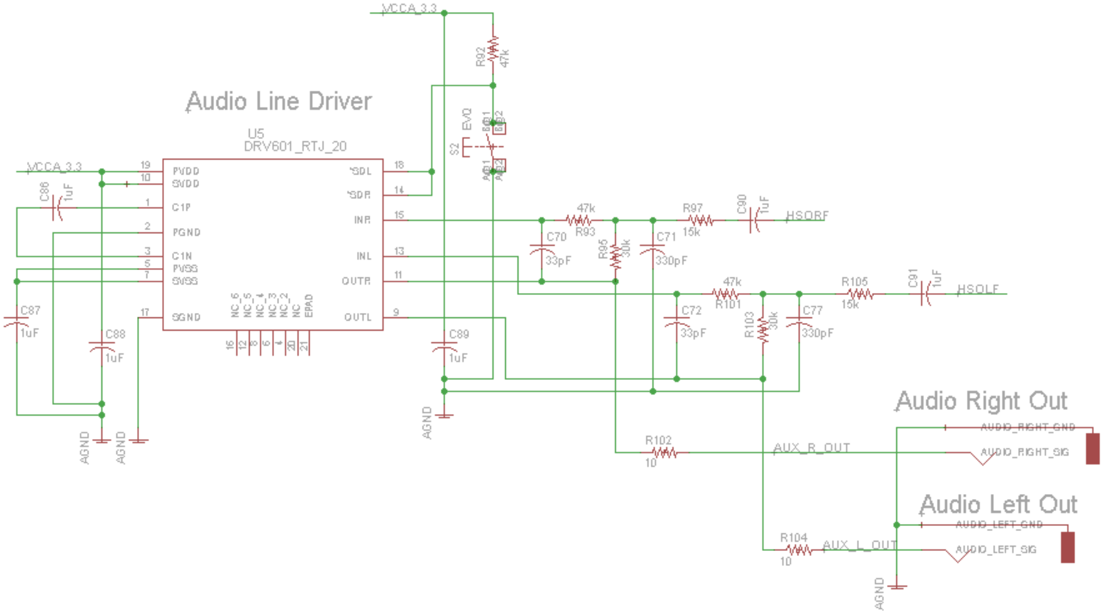


Figure 4 – 10: Audio Line Driver Schematic

4.10 Power Electronics

This section will discuss the intended power integrated circuits and buck/boost principles that Flashback will implore. This section will also discuss power layout and efficiency, along with the noise reduction design Flashback requires, since noise may disrupt signals

4.10.1 AC – DC Converter

The AC – DC converter Flashback used is 6V/2.5A power supply. It is a transformerless switch mode power supply (“Transformerless switch mode power supply”). It has a wide range input voltage, 85 V to 250 V AC, which is perfect seeing how the input voltage projected will be around 120 V AC. The Flashback board will accept this DC voltage input through a PJ1-022-SMT CUI, Inc. power jack. This power jack is rated to withstand up to 16 Vdc and 2.5 A. Although, the input supply current and the max input current are the same we have witnessed zero damage when testing the connector. ("PJ1-022-SMT").

4.10.2 Power ICs

With a DC input of 6V we needed three different output voltages 1.8 V, 3.3 V, and 5 V. The 3.3V will fill an entire power plane and the other 1.8V and 5V will be used through traces.

To provide these necessary voltage rails, our board starts the step-down process through two TPS6211x family ICs run in parallel. The TPS 62111 will be used to step down the 6V to 3.3V plane and the TPS62112 will be used to step down the 6V to 5V. The TPS6211x parts are able to accept 3.1V to 17V input range and output an adjustable voltage range from 1.2V to 16V with fixed output voltage options available in 3.3V (TPS62111) and 5V (TPS62112). Both of these ICs can output up to 1.5A current and the quiescent current is a mere 20µA. In troubleshooting these circuits the overcurrent and overtemperature protection came in handy because we had a short on 3.3V to GND which caused both overcurrent and overtemperature ("17-V, 1.5A, SYNCHRONOUS STEP-DOWN CONVERTER"). Please consult the schematic figure x-xx for how we implemented the input supply to 3.3V and 5V.

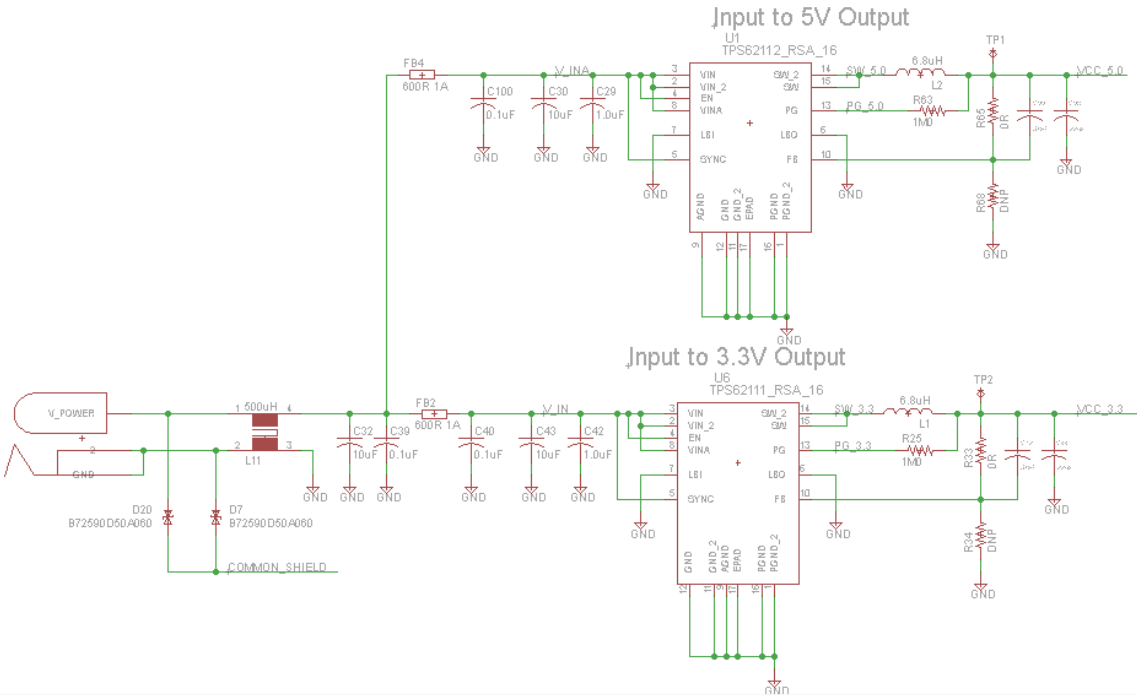


Figure 4 – 11: Power ICs Schematic

From our 3.3V plane, Flashback will use two MIC5247 efficient and precise low-voltage linear regulators to step-down the voltage to 1.8V. These products offer ultra-low noise, 85µA quiescent current, 150mA output current, thermal shutdown, and has a fast transient response (“MIC5247”). There are only a handful of items on our PCB that require 1.8V. Please see the schematic figure 4-12 for how we implemented the 3.3V to 1.8V step-down.

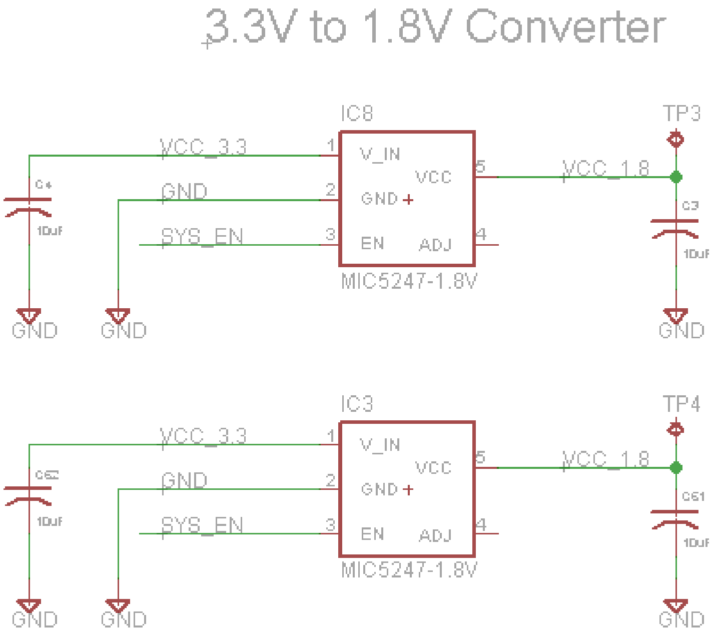


Figure 4 – 12: MIC5247 Schematic

To provide the proper amount of current and power enable for our multiple USB ports we'll use two ICs, the EP5368 for hub power and the TPS2052B for power enable. EP5368 from ENPIRION is a synchronous buck converter that can translate 2.4V to 5.5V into 3.3V at 700mA output ("EP5368QI"). TPS2052B is a Texas Instruments power distribution switch where the output follows the input at 500mA current ("CURRENT-LIMITED, POWER-DISTRIBUTION SWITCHES"). These devices are implemented in the following schematic figure x-xx.

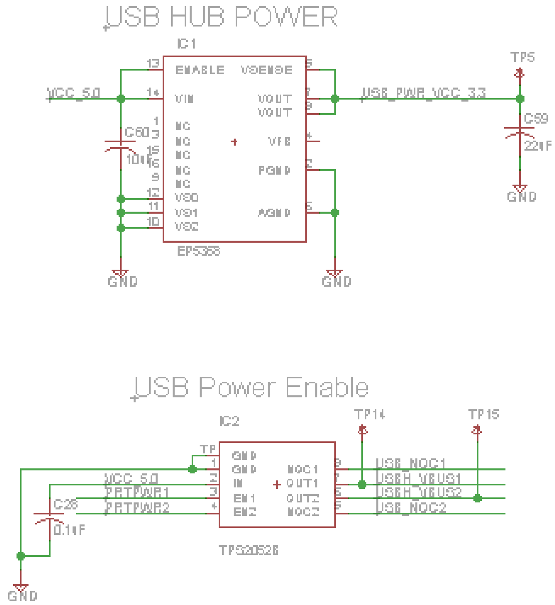


Figure 4 – 13: USB Power Schematic

4.11 PCB

Flashback utilizes a 6 layer board designed using EAGLE CAD, as shown in Figure 4-14. Using Eagle, along with a wide multitude of parts, provided many difficulties for this group. First and foremost, we had a learning curve issue. As none of us in this group has had any experience with Eagle, let alone PCB layout, it took some time to get in the flow of the layout. Another issue was finding libraries for all the parts we needed to provide the essential functionality of Flashback. We have a wide set of peripherals and at least one IC for each port. The board we designed contains 10/100 Ethernet, two USB-A 2.0, USB Mini-B, DVI-D, composite video, right and left audio channels, power supply, 7-segment display over 5 pin header, Overo TidalSTORM connectors, 3 pin header for a fan, and an infrared sensor.

By making a priority of list with time and data sensitive traces as our priority, we were able to route the DVI-D, USB ports, and Ethernet first. All differential data traces were routed as closely as possible to avoid any discrepancies that might arise if one of the signals needed to go around a resistor while the other was a straight shot. After all data lines were laid out on the board, we focused on making sure all the devices had proper power rails. As discussed in the power section, we needed three different operating voltages 1.8V, 3.3V, and 5V. Layer five was dedicated strictly to 3.3V. Except for layer two and five the 1.8V or 5V rails could of been placed on any layer. Our board also uses our ground layer, layer two, for thermal dissipation. By allowing thermal vias access to 13.8 sq. inches of real estate, the heat of an IC will dissipate across the plane.

Our board also has analog and digital planes. To accomplish the separation, we placed all components needing an analog ground and voltage plane together. Then we created a separation of 50 mils and bridged the gap to complete the current loop using 0Ω resistors.

Due to the amount of components on the board and the surface mount pads used for the Overo connectors, many of our data traces were six mils in width, power traces were at least eight mils, and the vast majority of vias were eight mils. Our board contained nearly 1036 surface mount pads, 252 different components, and 13.8 sq. inches with components on both top and bottom.

We sent our board files off to Sunstone to be printed because they were relatively the cheapest, quick turnaround time, and they were able to work directly from our Eagle files so we didn’t have to spend time converting to Gerber files. After the board was printed by Sunstone, Quality Manufacturing Services, Inc. in Lake Mary, Florida helped out our team tremendously. They assembled our board, helped us troubleshoot, and performed the rework on necessary areas of the board.

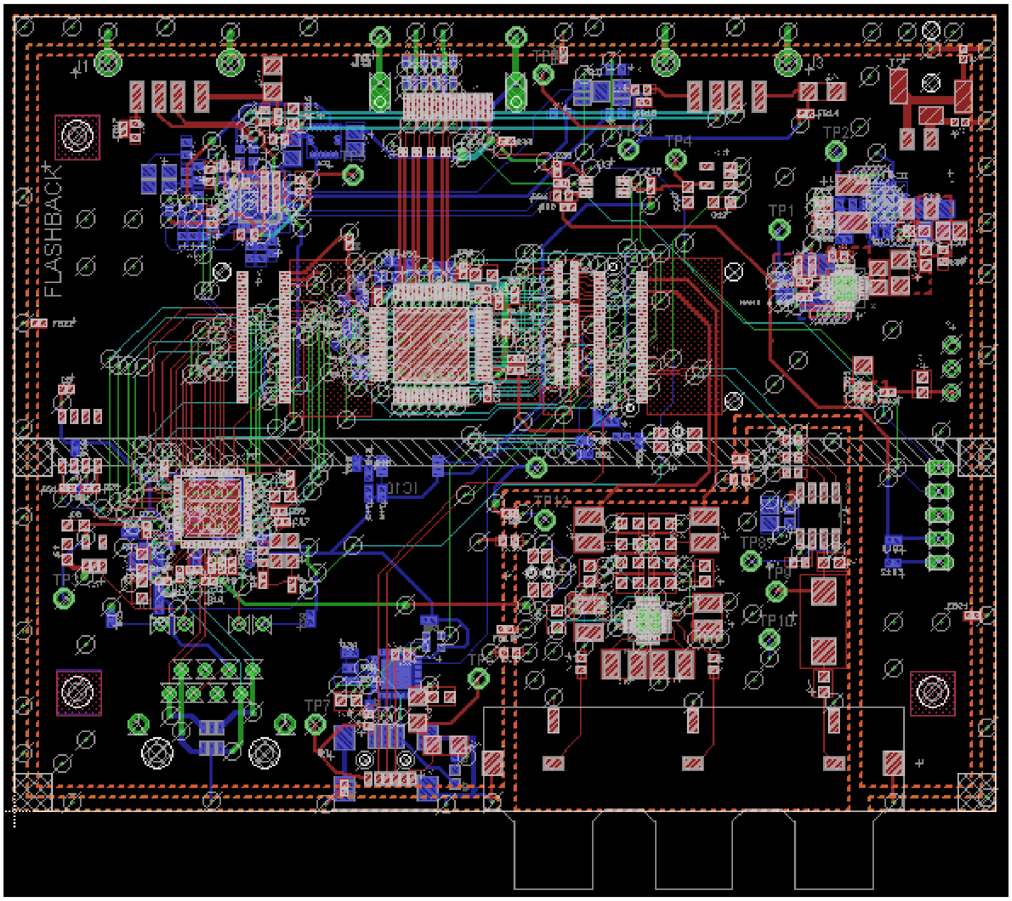


Figure 4 – 14: PCB

4.12 Debug

Flashback uses two different ports to debug the system, USB Mini-B and Ethernet. The USB Mini-B must communicate with the processor over UART due to pin availability on the Overo connectors. We used the Future Technology Devices International Ltd. FT232RQ USB to UART IC. The IC is set for USB 2.0 full speed and contains all the necessary USB protocols on chip. We used this port for reading bootloader and kernel loading output for initial system setup and debugging ("FT232R USB UART I.C."). In addition, it provided real-time control over our system. The schematic Figure X-XX below shows the schematic implementation of the port and USB to UART IC.

Ethernet will provide another means of debugging. We will use Ethernet for quick debugging, access to drivers/software packages to extend our operating system, update software packages, and quick file transfer after cross compilation in our build environment. Since there is no LAN chip on the TidalSTORM, we implemented our own. We used the LAN9221 which as an integrated Ethernet PHY with HP Auto-MDIX support, many reduced power modes, single chip Ethernet controller, and a host bus interface ("High-Performance 16-bit Non-PCI 10/100 Ethernet Controller with Variable Voltage I/O "). The schematic figure X-XX below shows the schematic implementation of the port and LAN chip.

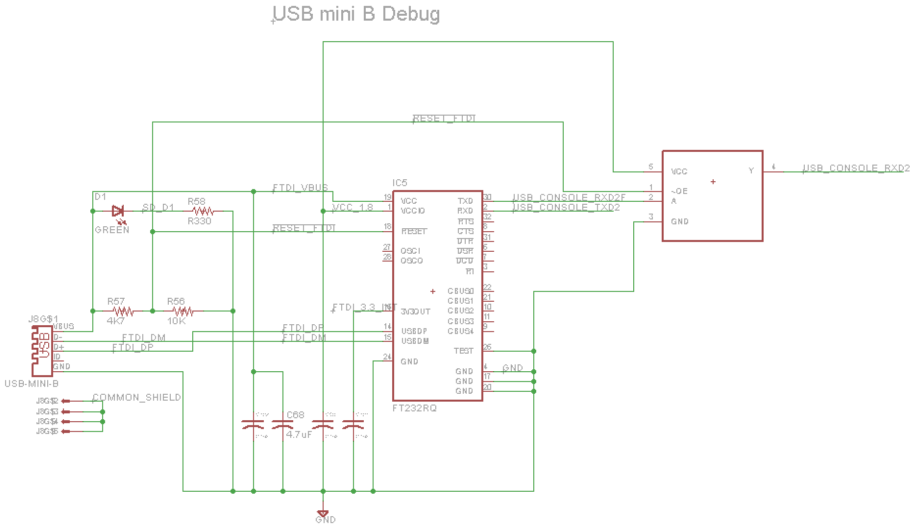


Figure 4 –15: USB Mini Schematic

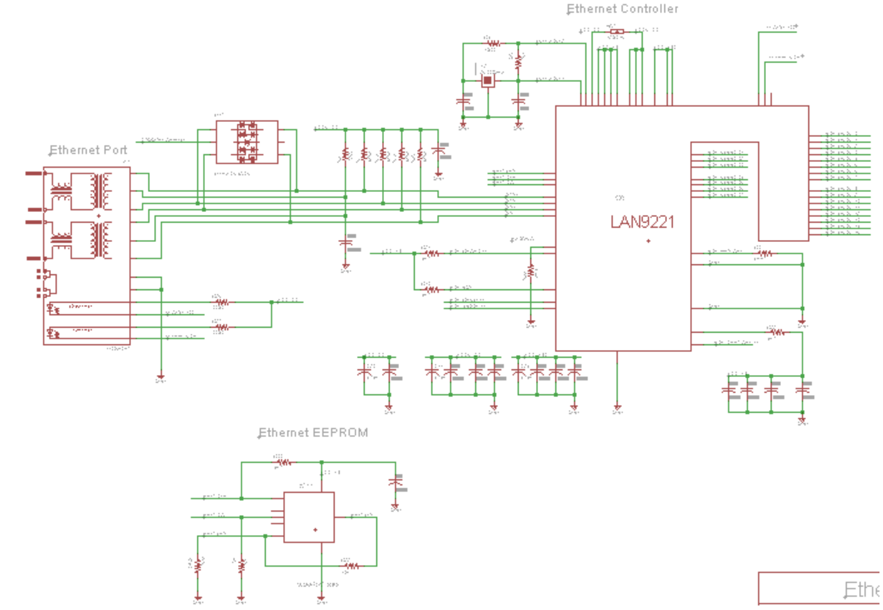


Figure 4 – 16: Ethernet Schematic

4.13 Operating System

After completing our research on viable operating systems, our group has decided to use a distribution of Linux that has been tailored for ARM architecture and the processors that support the architecture. This operating system is known as ArchLinux ARM and is widely used as a light weight solution to mobile systems that need powerful or high level software capabilities of an operating system and the simplicity and speed of a real time scheduler. Aside from the aforementioned features, ArchLinux ARM has other capabilities that made it appealing. Some of these features are as follows:

- Actively maintained
- Supported by the Arch User Repository – this means that aside from the packages that are officially released by ArchLinux, an ArchLinux user has the ability to develop, post and download user packages.
- Based on Unix – Our group is familiar with unix commands
- High level language support – this will make it very easy for us to develop in C++.
- Lightweight and flexible – we won't have to dedicate a lot of system resources such as RAM and processing power to running the operating system.

Our group concludes that ArchLinux ARM will fit our use case very well.

5.0 Design Summary: Hardware and Software

This section will discuss the software designs, such as commercial detection algorithms and the user interface, as well as the hardware designs, like the signal processing and output expectations. Flashback's logic design will also be discussed in this section in more detail than previous sections.

5.1 Graphical User Interface and Overlay

Our graphical user interface provides users with a familiar user experience that is easy to use. The menus are all non-invasive allowing users to adjust settings without missing parts of the show they are watching. The GUI provides a side menu, a slider ticker bar, and a notifications widget. The side menu contains all of the functionality the Flashback algorithm sports which are; disable/enable Flashback, select primary and secondary channels, view recordings, and set recording options. Of the four commands listed, the channel selection, view recordings, and set record options all have their own widgets associated with them which provide more options of users to access.

When users wish to change the channels they wish to view and run Flashback on, we've implemented a widget that allows users to view what the current channel numbers for their primary and secondary along with arrows indicating that the channels can be selected by selecting the up or down button on their remotes. The widget's functionality is still in the developmental phase at the end of this project but will allow for easy incorporation of remote control functionality in later revisions of the hardware and software.

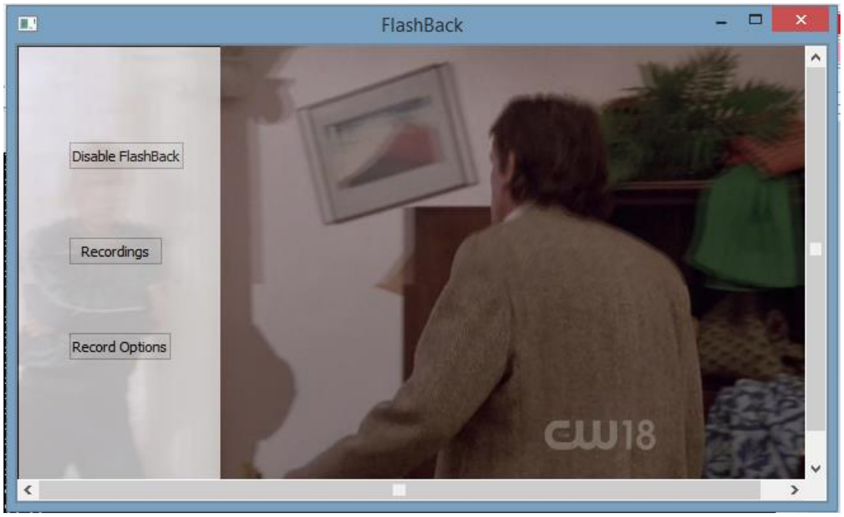
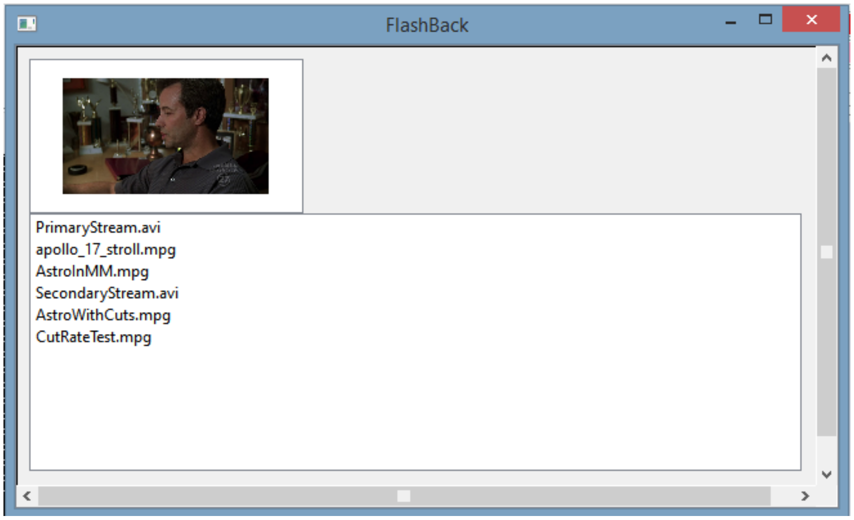


Figure 5 – 1: GUI Side Menu

For users to access their saved programs we implemented a list view window for which to display all of the recordings within the backing store which can be accessed by selecting the recordings option from the side menu. Since this menu takes up the entire screen, we've provided the user with a picture in picture view of the currently streaming video in case they are still interested in what is streaming before they make or cancel viewing a recording. When a user selects to exit this window or view a recording, the widget returns to streaming video in the dimension that the widget is set to.



PrimaryStream.avi
apollo_17_stroll.mpg
AstroInMM.mpg
SecondaryStream.avi
AstroWithCuts.mpg
CutRateTest.mpg

Figure 5 – 2: Recordings List View Widget

To give control to what the user desires to record, the record options widget allows users to decide which stream they would like to record. The widget is lined on the top part of the GUI and is also minimally invasive. The widget contains three different buttons; Record/Cancel Primary, Record/Cancel Secondary, and Cancel All Recordings. When a stream is set the record, the button associated with the stream is updated to show allow the user to visually see which streams are being recorded. As an added feature to the recording option, if the user has selected to have the Flashback algorithm analyze the video streams then the recorder will omit any commercials detected.

The slider visual widget shows the current placement in the hour the user is watching the show in and also contains a pause/play button. The functionality of these components hasn't been developed but with the graphical legwork complete along with the creation of the proper functions for which typical media player actions can be developed into the application.

The last of the widgets this application sports is the notification widget. This widget contains the same characteristics of the other widgets where it is non-invasive. The widget itself simply returns information to the user regarding the status of the channels they are tuned to. When Flashback has detected that the primary or secondary channel have returned from a commercial break, the notification states which stream has returned from the break along with a time stamp of when the detection was determined.

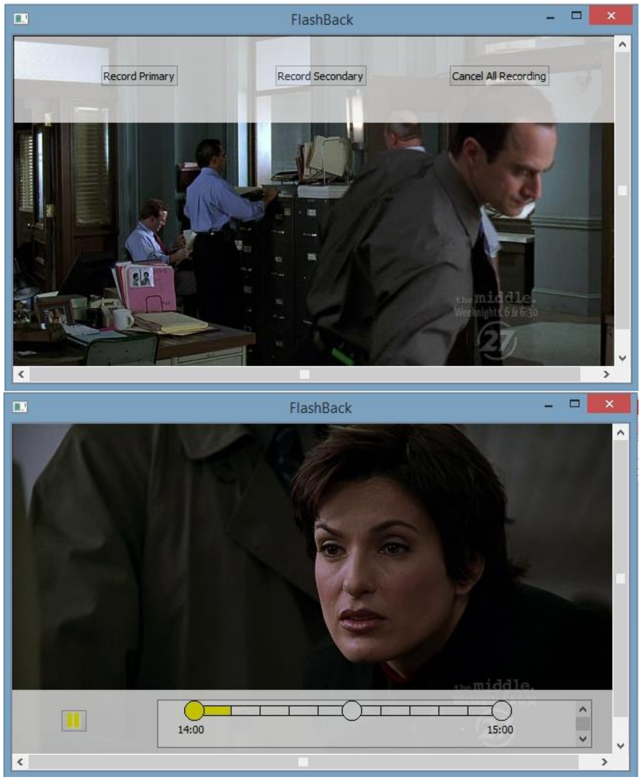


Figure 5 – 3: Recording Options Menu

5.2 Flashback Algorithm Analysis

The main function that Flashback needs to be efficient at is commercial detection. Upon researching into various articles of different possible algorithms that can be used to detect commercials, a black screen accompanied by silence was the most common solution to this problem. However the ability to detect commercials with a high rate of success requires more than just detecting for a black screen and silence. “Such simple approaches will fail for videos of TV channels that don’t use black frames to flag commercial breaks. Also, black frames used in other parts of the broadcast will cause false alarms” (Pinar Duygulu, Ming-yu Chen, Alexander Hauptmann). In order to detect these commercials effectively, we’ve used a series of triggers that will fire off an answer and when the answers for these triggers align we will than determine that a channel is on or off commercial.

Different commercial detection algorithms come with different strengths and weaknesses but when combined together in an efficient way, they can become powerful and effective. The algorithms in which we’ve implemented are black screen detection , high cut detection or high activity detection, and channel logo absence detection. Any one of these previously stated triggers have a tendency to appear in regular programming which inhibits the use of them individually, but when they are used in sync with at least one other trigger the recall ratio and precision are increased since the system is more robust.

Throughout the implementation of whichever algorithm chosen, we want a combination that provides a high recall ratio and high precision. The recall of an algorithm determines its effectiveness in detecting advertisements in relation to the actual programs length, while precision determines an algorithm’s strength in detecting individual commercials during the scheduled program. The equations that yield an algorithms detection strength contain the following operands; True positives, false positives, and false negatives. Let true positives equal the total number of commercials within a commercial block, let false negatives equal the total number of commercials, and let false negatives equal the total number of frames the algorithm detected as a commercial but was actually part of the desired program. Using these variables we can derive these equations; “Recall = (TP-Fn)/TP, Precision = TP/(TP + FP)” (Dimitrova, Jeannin, Nesvadba, McGee, Agnihotri, Mekenkamp).

Our implementation of black screen detection was fairly straight forward as it mostly involved addition as its main operator. The software would go through a given sample space of frames and sum up each frame within that sample space. In an ideal world, any given sample’s sum should return zero if it is a purely black frame, but since frames all contain noise the sum of a black frame is significantly larger than zero. The threshold of this trigger is set at 200000; you may be saying that this number is significantly large for a threshold but a typical frame with different palates of color can have the sum of upwards of two million, so this threshold is significantly low in reference to typical frame values. After this algorithm has been implemented, it has never missed a black frame detection further stating why this attribute is one of the most simple and effective algorithms.

High cut rate detection is a little more complicated than black screen detection. Cut rates are how quickly shots change in a selected period of time. Naturally commercials are designed to advertise as much information as possible in a short amount of time which creates a high cut rate. The way to determine a commercials cut rate is by determining the program’s standard deviation in pixel densities and comparing a frame’s pixel density to that deviation. Once the algorithm determines that a frame is outside of the deviation, the algorithm can then begin to count the number of frames in which the activity is different than the deviation. The number of frames detected with high activity should theoretically be the span of time in which the program was on a commercial break. Similar to black screen and silence detection, this trigger cannot be the only trigger in an algorithm. In some cases, programs will have high activity which will cause false negatives and parts of the program may be skipped.

The last trigger we’ve implemented logo absence detection. During every commercial break, a channel’s logo disappears letting us as viewers know that the next frames will be advertisements. This algorithm requires two outside variables to be effective; a sample of the logo matted on a black background and the location of the logo in any given frame. When the algorithm obtains these variables, it employs an edge detection technique to compare the sample of the logo with no noise to the sample of the logo obtained for analytics. The threshold of this algorithm is set to be the sum of the pixel values in the sample logo with no noise divided by three. We chose this value purely by observation of the differences between the main sample of the logo and the sample obtained for analysis.

Our device also offers recording television without commercials. The diagram below accurately depicts the process in which our recorder omits commercials from the desired station. Each stream has a Boolean associated with it in regards to its status and based on that Boolean the recorder for each stream is given frames or not.

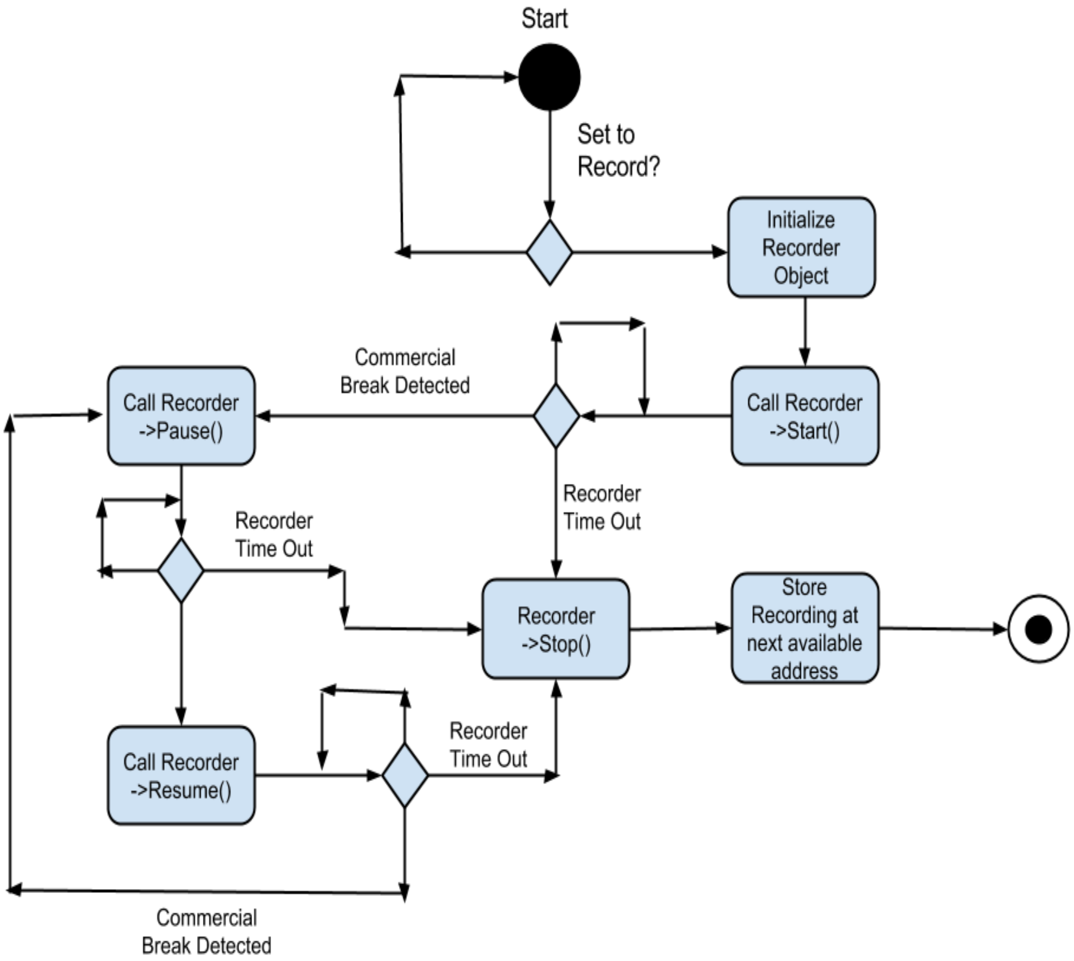


Figure 5 – 4: Recorder Lifecycle

5.3 Hardware System Overview

Flashback device will be able to receive broadcast stations from three different input signals. The first being antenna (OTA) transmissions. Flashback will first focus on proving the concept through OTA television signals because of the encryption factors that come along with cable and satellite based devices. Our system will account for all of these features for future implementation. If our device is able to implement all three of these features then we will send the signals to a switch to determine where our incoming signal will originate from. The incoming signal will navigate to our device through a coaxial cable medium. Flashback will have the necessary port to receive the coaxial signal and distribute the signal to the necessary devices.

After receiving the incoming signal from broadcast stations, we will employ a multi-channel hybrid tuner. The hybrid tuner we have chosen can process digital and analog signals of multiple different standards. It is a worldwide device developed from Silicon Labs (part number: Si2157). This television tuner will communicate with our OMAP4470 processor package via I²C protocol. This tuner will be able to receive signals from the OMAP4470 (which received input from the user) to select certain channels for the Flashback feature and viewing.

Upon receiving the television signals from the Dual TV tuner the OMAP will communicate with memory by means of a SATA driver running up to 3.0 Gbps. The OMAP will tell the memory to store at least two video channels in memory. By storing these signals in memory, we enable the device to analyze the incoming signals for commercial breaks, rewind, fast forward (after rewinding), and pausing the program on the television screen. The memory devices could also contain necessary code for Flashback feature and GUI.

After the television signal has been processed by OMAP, the OMAP will send video signals through either a digital to analog converter or through HDMI. The video DAC (part number: AD5629R) will be used for the standard definition television signals. In addition, the DAC will also be the same DAC to process the corresponding audio signal for the television program. Both the Video and Audio will need to pass through a filter or line driver. From there these signals can be forwarded to the television display unit (video signal) and the audio interface whether that is the television or a speaker system. The HDMI signal will actually contain both the video and audio signals. Due to the high speeds and sensitivity to changes in voltage, the HDMI port will need to pass through an electro-static discharge protection circuit. After the ESD circuit, the HDMI signal will communicate with the television unit.

All of the decisions for channel selection, flashback enable, volume, GUI will come from a user interface module. This user interface module will contain the on-device keypad, remote control and it's corresponding receiver in Flashback device, and seven segment display to show the user what they have chosen.

To monitor the devices heat dissipation, we will implement a temperature module. In this temperature module we will use an analog temperature sensor integrated circuit. The integrated circuit approach provides consistent data as temperature increases or decreases within -50° to 150° C. The change in voltage as temperature increases will create an analog waveform. By sending this waveform through an analog to digital converter and then to the OMAP we can interpret this waveform and determine if the system needs to shut down to "cool off." There will be more than one temperature sensor IC on the device. We will place them at strategic locations around the OMAP to monitor not only the OMAP but also the heat coming from peripherals and power electronics. To help regulate the temperature our system will employ heat sinks and a fan to increase air flow within the device.

Flashback's power module will take in a 120 V AC source from a US standard wall outlet and convert it down to 12 V DC. From the 12 V DC we will use LDOs to get the correct power rails for the rest of the board components and FET load switches to monitor the power signals.

The OMAP processor will handle the workload of analyzing the incoming television signals, relaying them to video and audio interfaces, communicating with internal LPDDR2 RAM, solid state drive, user interface, temperature module, and this device will be receive power from the power module.

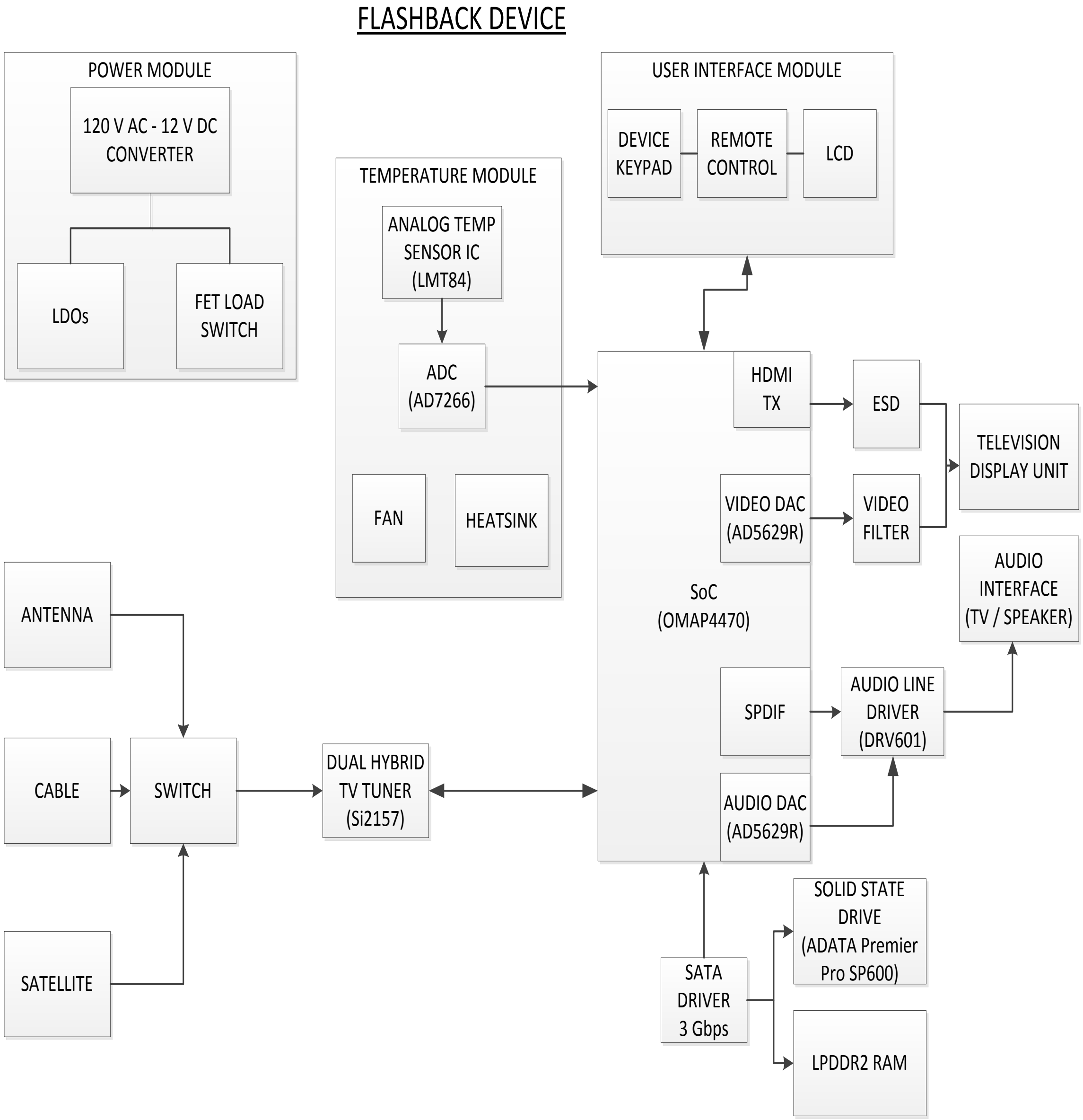


Figure 5 – 5: Flashback System Function Block Diagram

6.0 Project Prototype Construction and Coding

This section will discuss the "plan of attack" for Flashback, including what testing and prototype application will be used in order to achieve Flashback's full potential

6.1 Software Development Environment

Our group has done research on two Integrated Development Environments, (I.D.E.) known as Microsoft Visual Studio and QtCreator. As mentioned before, we will be using a C++ Graphical User Interface Framework and library called Qt. To develop the front end of Flashback using Qt we will need one of these IDE's.

Our group has decided to use QtCreator. We decided upon QtCreator because we expect it to better support the Qt framework since it was built specifically to support graphical user interface development. Our other option, Microsoft Visual Studio can work but will require some steps to set up support for Qt. Once it is ready to be used for development we are limited to developing and testing on machines that have Windows 7 or higher installed. Since our group is using a Linux distribution to provide high level programming language support, mainly for Qt and C++, we thought that using Qt will be better. Because Qt can also be ran on most Linux distributions and thus will allow us to compile and debug on a Linux distribution before testing on our target hardware. We need this capability because it will speed up the debugging process greatly and allow us to temporarily circumvent the other option of side loading the Flashback software every time we make a bug fix or significant update, when we want to test that our software still meets our initial specifications.

6.2 Graphical User Interface Overlay Plan

We created a GUI overlay that seamlessly streams the channel that the user has selected and contains menus that are functional and easy to use. The Qt framework provides a variety of widget libraries for which we can design our own overlay at ease. The classes that will have the viewer see their program is QGraphicsView. QGraphicsView in itself is actually a container class used to host a QGraphicsScene which, in turn, hosts all of the possible graphics items that will be designed by us. The QGraphicsView is also a layout item within Qt Creator's layout form; using QtCreator's window builder, we can visually adjust the size and placement of the QGraphicsView we are seeking to implement onto the on screen display.

QGraphicsItems are used to fill a QGraphicsScene, these graphics items are fully customizable and can contain their own widget properties and actions. Taking full advantage of the QGraphicsItem class we can create widgets specific to

Flashback. Within Qt there's a special QGraphicsItem called QGraphicsVideoItem. This item is the key to streaming video files and will be used to stream the MPEG-2 compressed files that will be compressed after they're decoded by the TV tuner. Figure 6 – 1 shows an example of streaming video through this graphics item. With this main item in place we can create the rest of our items accordingly. These interfaces are primary and secondary channel selection, a general options screen, a layout for each recording option, a previously recorded program accessibility screen, and pre-pivot notifications for the user. These interfaces are illustrated on Figures 6 – 1 to Figures 6 – 10.

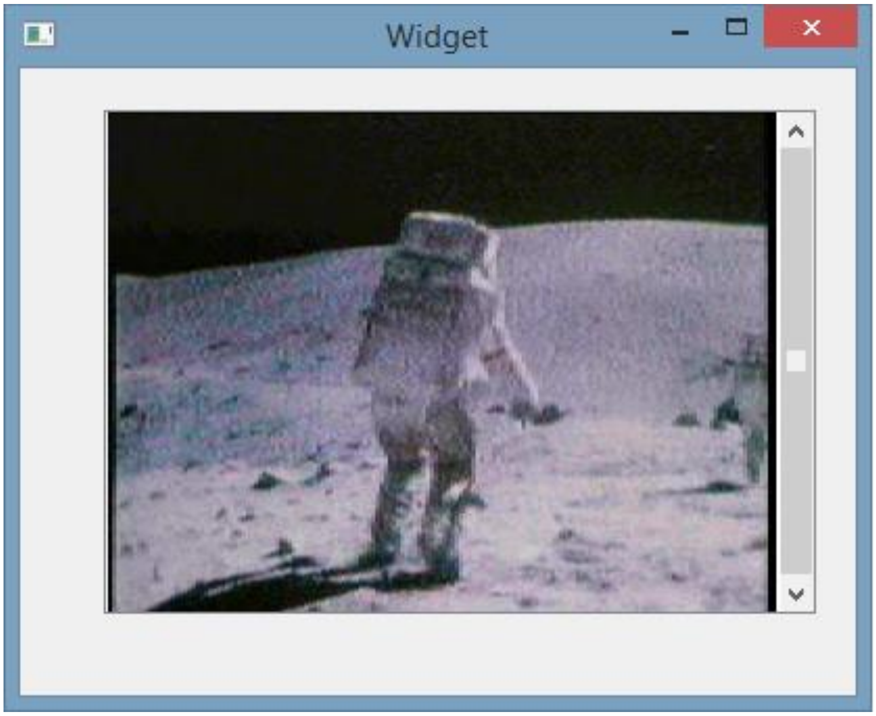


Figure 6 – 1: Streaming Video from Qt Application

Some of the menus we plan to include will have two possible layouts in which the user can adjust their settings. One view that a user can select is for a long and skinny menu on the left side of the screen, the other view a user can select for their menu is full screen. The 'taskbar' like menu setting will contain less information about the options for which users can set, however this option allows for a transparent menu bar that doesn't block the user view from their current programming. This view's intention will be to provide users who are familiar with the settings in that menu with an easy way to access their preferences without having to pause or rewind what they're watching. The full screen menu will contain more information about each possible setting a user can select and adjust. When accessing a menu that is using the full screen layout, the streaming video, and recording if applicable, will be paused. The menu's current design does not contain a picture in picture element that can stream the video playing and will not contain any transparent attributes; pausing what's currently playing will allow the user to change and apply settings without missing any portion what they are watching. This screen is intended for users who aren't familiar with the possible settings they can adjust.

Whether the user has selected for their type of menu to be the taskbar or full screen menu, the user will be able to see different menu items for which they can manipulate. The menu items for which a user can select will be; the option for menus to appear in the taskbar or full screen format, the option to record the primary or secondary channel onto the disk, the option to automatically switch between channels without notifying the user. Within the taskbar styled menu, these options will have icons associated with them, within the full screen layout these options will have the icon along with text describing what the option is along with a checkbox or button group to select what would be preferred by the user.

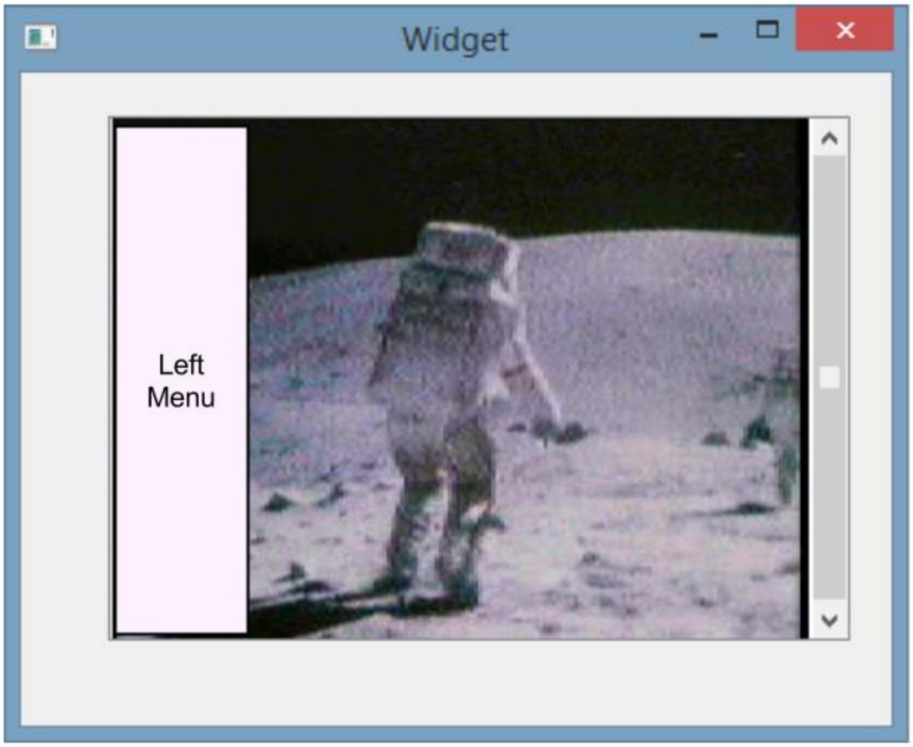


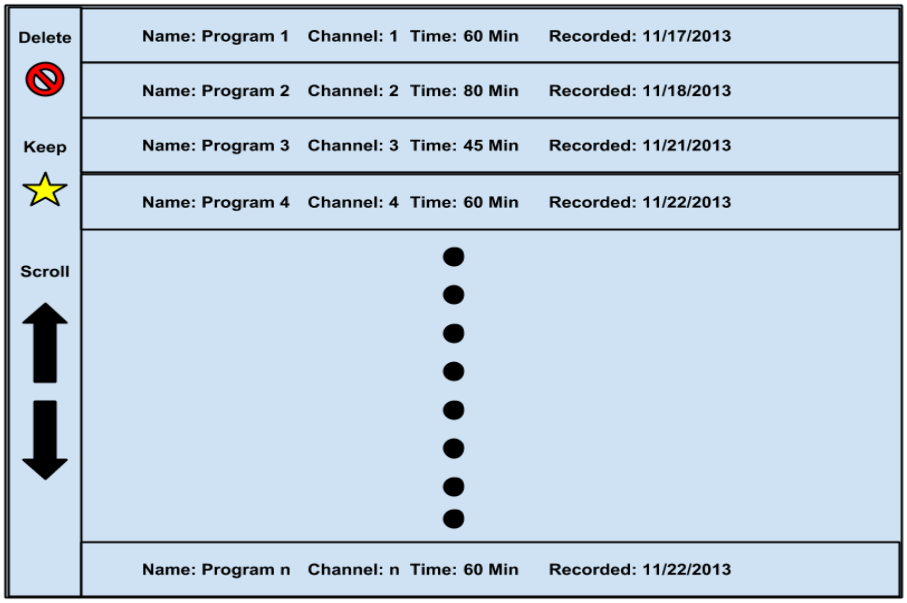
Figure 6 – 2: Task Bar Style Menu

Another GUI component will be the GUI the user will use to select which two channels the Flashback algorithm will pivot from, or Channel Pivot GUI. This GUI will be of medium size since not many widgets need to be added to this specific GUI. This specific GUI will contain labels, arrows, text boxes for the numbers, and a set button to confirm that the channels selected will be pivoted upon. The design for the Channel Pivot GUI is meant to be user friendly and easy to Figure out how to adjust channels and set them. The users will change the primary channels with the channel button and will change the secondary channels with the volume buttons.

If the user decides to set either their primary or secondary channel to record from the general options menu, the user will be prompted with another GUI to determine the length of the recording. Within the GUI there will be two options for which the user can select, a user can manually set the time in which they would like the recording to end and the other selection will be a selection by genre which will record based on the average time's that genre of program usually lasts in a televised setting. Every program being recorded will have an end time, if a user decides to avoid setting a time, recording will not begin. Both of these options will have their own distinct layout, the manual input for recording time will consist of a four spin boxes, one spin box for the hour of the start time and another spin box for the minute within that hour, the same will be for the end time section of the GUI. Next to each of these time specifications will be a AM or PM setting as well. Although users can manually set the recording time, they may make the mistake in recording for unreasonable hours, thus potentially eating up the memory available on the drive. To counteract this possibility, the maximum time a user can record any given program will be no more than the maximum average time of all the genres on the hash table. To represent the genre selecting portion of the GUI, we'll use another list view layout. This layout will contain the name of the genres along with the average time they record for.

To provide ease in replaying previously recorded programs, a list view widget will be designed and linked to the main GUI overlay. This list view GUI will contain a list a references of previously recorded programs that are on the drive. Many options such as delete, keep, and play will also be contained in the GUI. We also plan on implementing a multi-choice system so users can delete more than one program at a time. This GUI will be required to take up the entire screen due to the amount of information needed to describe each recorded program. Each list item will contain the name of the program, how long it is, when it was recorded, as well as which channel it was recorded on. If there are more recorded items than the actual GUI has space for, users will have the option to scroll down the view to see the programs that aren't shown on the screen. When a user selects a program to watch, the list view GUI will send the address of the requested recording to the background process, once the address has been found, the GUI will be closed and the graphics view will be seen. Once the list view GUI is killed, the recording will commence its playback.

User's will also have the options that a standard media player would contain such as fast forward, rewind, stop, and resume. When a user is viewing a recording, the advertisement detection algorithm will be stopped since there are no frames from a live signal to be analyzed and compared. When a user has completed watching the playback of the recording, the user will be prompted to either delete or keep the file.



Name: Program 1 Channel: 1 Time: 60 Min Recorded: 11/17/2013

Name: Program 2 Channel: 2 Time: 80 Min Recorded: 11/18/2013

Name: Program 3 Channel: 3 Time: 45 Min Recorded: 11/21/2013

Name: Program 4 Channel: 4 Time: 60 Min Recorded: 11/22/2013

Name: Program n Channel: n Time: 60 Min Recorded: 11/22/2013

Figure 6 – 3: List view GUI Layout

The final component we plan to add to the GUI is a popup notification widget to let the user know that the Flashback algorithm detected a station that has completed its commercial break. This notification widget will be a simple GUI containing mostly information and a button in which user can select to indicate that they wish to switch channels. The layout for this component will contain the name of the channel, the time it completed its commercial break, whether the channel is primary or secondary, and the actual flashback button for manually switching. If the user has selected for automatic switching, this component's button will not be clickable, but it will still warn the user that the channel is about to switch. This component will maintain transparent qualities so as to avoid interrupting the user's view of the program they are currently watching while having text that's legible and clear to read.

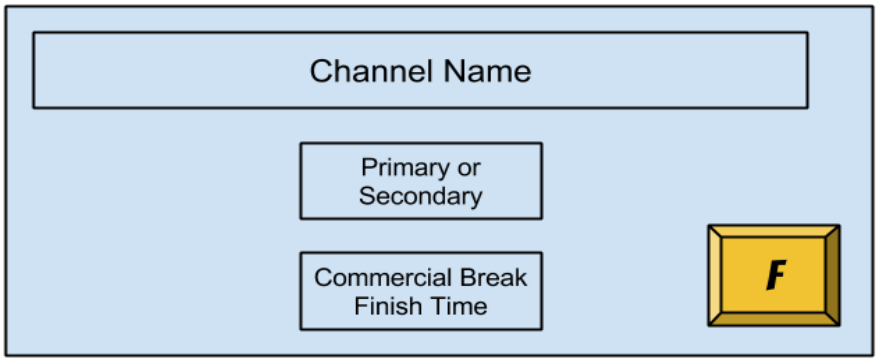


Figure 6 – 4: Notification GUI Layout

Following these design plans we can effectively create our Graphical User Interface with the ability to add whatever graphics to it that we wish while maintaining its functionality.

6.3 Flashback Hardware/Software Diagrams

Software Diagrams

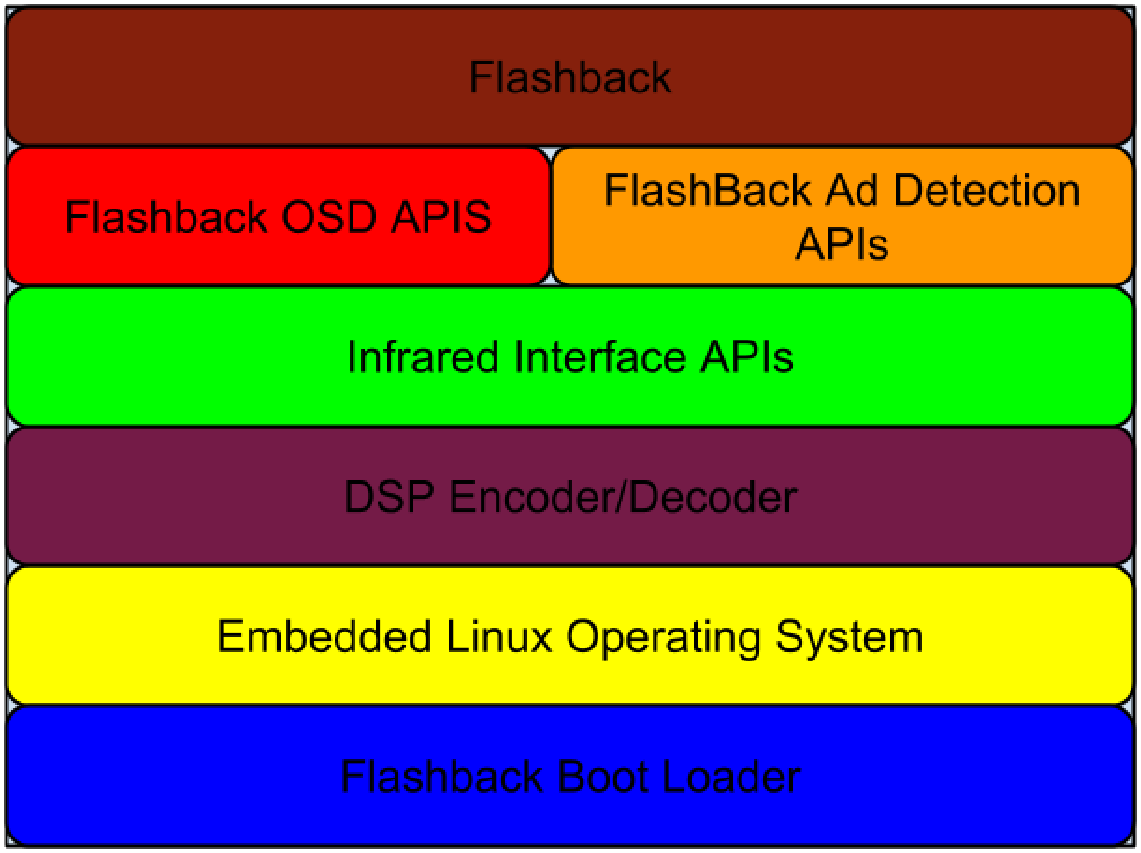


Figure 6 – 5: Flashback Software Layers

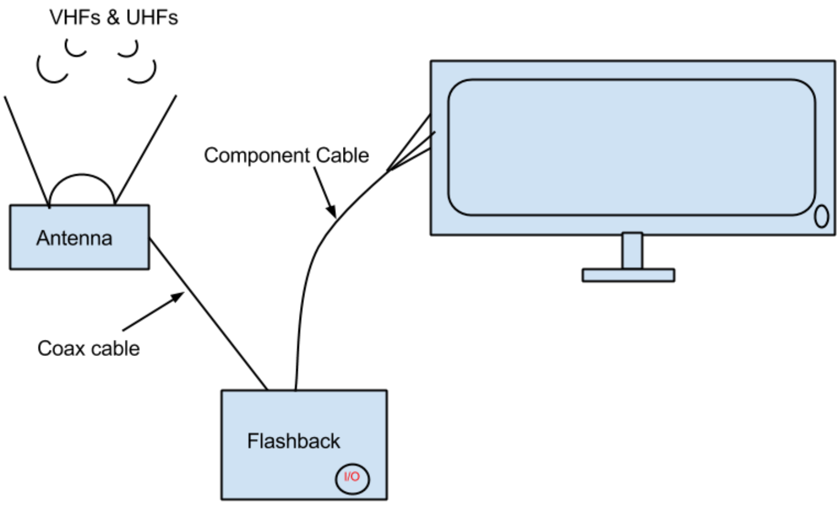


Figure 6 – 6: Projected Flashback setup

7.0 Project Prototype Testing

This section will discuss how Flashback's prototype and eventual first generation design will be tested, as well as what hardware and software benchmarks we intend to pursue and examine.

7.1 Software Benchmarks

This section will discuss the software standards that Flashback is intended to fulfill. It will discuss, in further detail, the expectation of the commercial detection performance and potential testing methods for timing targets

7.1.1 Commercial Detection Algorithm Efficiency

To test our algorithm for robustness and accuracy, we created a control sequence which contains ten different commercial breaks not exceeding six minutes per block among a two hour block of actual programming. These commercial breaks will be randomly placed within the programming to make sure the algorithm can react in real time to the transition between program and commercial.

In creating this three hour video of commercials and programming, we used video from a couple of different stations that have logos so the logo detection portion of our algorithm can be tested. Note that these logos will be saved in storage and will be included used in a hash map with each channel's number as the key and the returned value is the actual logo object with the appropriate fields that indicate placement, logo size, and color codes. The correct number of commercial breaks as well as the time each commercial break takes will be known thus allowing us to determine how accurate the algorithm is.

The efficiency of our algorithm will be determined by timing how fast it can analyze each frame in the controlled test. If we have an algorithm that has a very high recall ratio and precision but lacks in its effectiveness in completing the detection in real time than we'll have to go back to the drawing board. The same goes for an algorithm that reacts well in real time but detects and switches for an unacceptable number of false positives. Ideally, our algorithm needs to have a recall ratio above 90 percent and a precision around 90 % before we can stick to a version of an algorithm and increase the efficiency.

The objective of our control test is to, in a sense, break the algorithm with a test case and record the findings and conditions in which the algorithm didn't perform as expected. Our test will contain irregular places for which there will be commercials as well as irregular lengths of commercial blocks so the algorithm we create can be as dynamic as possible.

We will also implement a method for which our algorithm can count each specific commercial; this ability has to do with the final trigger of our algorithm which is high cut rate detection. If our algorithm is reasonably close in the count of commercials within a given commercial block, we can infer that the algorithm is solid and can thus be used in the final production of our device.

The environment for which this test will take place will be on two different platforms, the first platform will be a computer, and the second platform will be on the actual device. The reason for initial testing must take place on a computer is due to not only time constraints but because it would make an efficient use of the time in which the board for the device is not complete and taped out. Testing on a computer would give the algorithm the direction of development it would need before even having the device manufactured. Testing on the actual device is also very important. Once the device has completed manufacturing we must take note on what hardware the device can actual use, how much memory it contains, and how high of a frequency its process can run at.

Testing on the device would give us the edge we need to write the final version of our algorithm to run specifically on our device ensuring that our final product is functional and meets the specifications we've set for it. The testing of the algorithm itself also opens doors for testing the hardware's temperature sensors, as well as connection between other components of hardware that may need to be changed to ensure correct operation.

7.1.2 DVR Algorithm Efficiency

To test the efficiency of the actual digital video recording portion of our algorithm we will read in any given television program in which we can recognize the order of the actual programming before and after commercial breaks. We’ll be using the same controlled test created to test the advertisement detection algorithm to test the recording portion of the algorithm. Testing will occur in two different environments, the development environment and the target device. In order to make progress towards building the software to our specifications, a working model needs to be produced. This working model will be built in the development environment which will be a computer. When enough results return positive, we can then implement and test the recording algorithm on our target device.
Three distinct behaviors we’ll be seeking to observe are; how reactive the system is to user input while it’s recording, how effective the device runs the commercial detection algorithm while recording, and last but not least, how well the device recorded the user’s program. If our algorithm uses too much CPU time in the development environment than we can safely say that the algorithm will need reworking before it can be implemented onto the target device. If the algorithm doesn’t lock the processor from performing other tasks but doesn’t record the program correctly in the development environment, the software will need to be revisited before testing on the target device. Any other issues that are significant enough to hinder the correct results in the development environment can be assumed to occur on the target device, thus requiring more testing on the development environment before implementation onto the device.

When the recording portion of our algorithm works correctly on the development environment, we can then implement and test this portion of the algorithm on the target device. When testing on the target device, a log of all the events that occur within the device will be kept so a trace of code can be done. Using this log, we can determine what lines of code didn’t operate correctly on the device but operated correctly on the development environment. If the playback of the recording on the device doesn’t match the playback of the recording on the development environment, we can analyze and compare the log from the target device to the log from debugging the algorithm in the development environment.

There are two different options for users to select in order to record programs, manual and genre selection. The controlled test will be run on both of these options to determine how they affect the efficiency of the recording algorithm as well as how they perform in terms of accurately recording and storing a user’s program. The testing of each option, however, will have minor differences in order to simulate how they will be used in a real life scenario when a user won’t know the exact second in time that their program has begun.

We’ll begin manual testing by making note of which sections of the controlled test we wish to record. These sections will also include commercials along with scheduled programming. One of the sections that we’ll record will end with the scheduled programming while the other section will end on commercials. The reason to make sure the manual test's ending time lands on both of these sections is to determine the behavior it has when it ends on a commercial since the algorithm will be cutting out commercials. When the test's end time is during scheduled programming, this will determine how responsive the algorithm is to that end time. Testing on these types of sections will determine the responsiveness the algorithm has during non-ideal situations.

When a user selects a genre to base their recording off of, the actual recording will begin at the moment the signal is sent from the user's selection. The recording will last for the given average time that those type of programs last for. To test this option, we'll expect that the user has selected to record the program from the time it was aired. We'll take note of the section's beginning we wish to record to determine how responsive the option is to user input and compare the recording to how much content the program actually had. We'll also be looking for how well it cut out commercial breaks.

The testing of each individual option will determine which option records effectively, which option's interface is easier to use, and which option recorded more of the scheduled program. The best option will be the one that will be heavily built upon and re-tested so as to supply the user with the best possible viewing experience as well as ease of use with the box.
The best option will be the recommended options for users to use but will not eliminate the other option to provide users with a choice in which they wish to record their programs.

Testing both of these options for recording and commercial detection efficiency will give insight into what the algorithm in the background is doing. Determining which portions of the algorithm need to be improved, these tests must be run until the device performs with the best possible accuracy and speed.

7.2 Hardware Benchmarks

This section will discuss the Flashback's user interface and power hardware, and the approach of how they will be tested to confirm operating levels and performance for the final product.

7.2.1 User Interface

To test the user interface, we must start with the IR remote and receiver. Since we are using the MSP430 as the microcontroller, we can use the LEDs on the microcontroller as a confirmation for all the contact points. Our group will then implement the IR LED and a basic receiver, possibly with an LED output, to check if the receiver is working correctly at the desire range.

Once the basic receiver is set, we need to translate it to the Seven Segment LCD IC, in order to get the display working with the correct user input. After the display is correct, our group will assign the numbers to the channel frequencies, which will have the saved in the addresses to being the “Flashback algorithm”. The device controls will go through a similar process but not with the MSP430 Microcontroller Launch pad. The device controls will have to have their logic tested and then later put in to the actual Flashback algorithm.

Once these are working, we will implement the logic from the software test bench and make sure the hardware performs accordingly. Doing small test will help us troubleshoot and debug better, as compared to trying to mount the entire thing together and finding a small problem in a large scale.

7.2.2. Power Electronics

Since one needs to be certified to create/replicate an AC – DC converter, our group has decided to simply buy one with a 12 V DC output. After measure every element, such as resistors, capacitors, and inductors, to ensure that they are the right ones used. After going through ESD protocol, described earlier, the two LDO connections to the AC – DC converter will drop the voltage from 12 V to 5V and 3.3 V. After measuring this voltage and the output rush current, we will try to implement every design possibility discussed to make sure the current does not affect the elements. Our goal is to make the ESD and FET Load Switch a outlier function, not a normal one.

Separately, referring to the data sheets, we will test the components, with simple circuits and temperature measurements to ensure that they are indeed working properly. Our group decided to order extra parts for emergencies, such as the chip breaking or getting lost, and for testing, in case we wish to test multiple ICs side – by – side.

For the ESD – sensitive elements, we will consider the maximum allowed temperature and current and keep note of that when measuring the power rails and relays used. The main thing that needs to be tested is the steady state in terms of time. We will let the circuit go through many clock cycles and use an oscilloscope to test the wavelength to make sure it is not fluctuating.

8.0 Administrative Content

This section will show the project timeline and intended dates of completion along with the projected costs for Flashback.

8.1 Project Milestones

(**Beginning Fall 2013)**

09/09	Project Definition Completed/Begin Research
09/23	Begin Part Collection/Begin Software/Hardware Design
10/14	Continue Main Research (Designation based on individual responsibilities)
11/7	Submit rough draft of Senior Design I document
12/2	Submit final draft of Senior Design I document
12/11	Software Design Completed/Hardware Design Complete/Implementation Begins
12/13	Part Orders and Collection Completed/Continual Hardware/Software Design-Basic Box Design/Begin Power Supply
12/24	Receiver Design and Sync/Power Supply Testing/Signal Translation
01/17	Implementation Completed/Simulations Begin
01/18 – 02/01	TROUBLESHOOTING: Software Simulations/Signal Analysis
02/02	Simulations Complete, Gen 1 Hardware Design
02/09	LCD Overlay Universalized/Being Prototype Design
02/17	Prototype completed/Begin Gen 2 Design
03/10	Software and Hardware FULLY Completed…Final Testing Begins
03/17 - 04/01	TROUBLESHOOTING: Testing
04/02	Testing Completed/Begin Preparation for Senior Design Evaluation
04/09	Completed Product/Preparation for Senior Design Evaluation

8.2 Final Budget and Cost Report

Product	Description	Quantity	Price (per unit)	Total Price
Overo® TidalStorm	DM3730 Development Board	1	$139.00	$139.00
Summit	Prototype Board	1	$49.00	$49.00
AVerTV H826	TV Tuner	2	$44.99	$89.98
74AHC1G126DBVRE4	Buffer and Line Driver	4	$0.28	$1.12
DRV601	Audio Line Driver	1	FREE	FREE
USBLC6-2P6	High Speed Interface Protection	2	$0.92	$1.84
DLP0NSN900HL2L	Common Mode Filters	12	$0.44	$5.28
DVIULC6-4SC6	ESD Protection	4	$1.26	$5.04
MC1411B	7 Segment LCD	2	$0.35	$0.70
REG711	DC-DC Converter	2	FREE	FREE
Connectors	Various	8	$50	$50
PCA9306	I^{2}C Bus	1	FREE	FREE
TPS2051B	Power Distribution Switch	1	FREE	FREE
THS7315	Video Amplifier	1	FREE	FREE
TFP410	DVI Transmitter	1	FREE	FREE
TPS62111	Step-Down Converter	1	FREE	FREE
RLC	Various	TBD	$20	$20
Shipping		TBD	$100	$100
PCB	6 Layers	1	$220.50	$220.50
Total	$682.46			
Budget	$1000			

* *May be donated*

** *Product involves multiple testing so exact number is not predictable*

Appendix A

Works Cited

1. "17-V, 1.5A, SYNCHRONOUS STEP-DOWN CONVERTER" Texas Instruments. Web. 27 Apr. 2014. <http://www.ti.com/lit/ds/symlink/tps62110.pdf>.

2. "3-Channel SDTV Video Amplifier with 5th-Order Filters and 5.2-V/V Gain " Texas Instruments. Web. 27 Apr. 2014. <http://www.ti.com/lit/ds/slos532/slos532.pdf>.

3. "About SATA Hard Drives and Controller Modes." *HP Consumer Support*. Hewlett-Packard, n.d. Web. 28 Nov. 2013. <http://h10025.www1.hp.com/ewfrf/wc/document?cc=us&lc=en&docname=c02961221>.

4. "Autohop." *Dish Network*. Dish Network. Web. 30 Nov 2013. <http://godish.com/hopper/autohop.asp&xgt;.

5. "BCD-To-Seven-Segment Latch/Decoder/Driver."*sentax.com*. Motorola. Web. 1 Dec 2013. <http://www.sentex.ca/~mec1995/tutorial/7seg/mc14511.pdf>.

6. "Choosing a DSP Processor." (n.d.): n. pag. *Berkeley Design Technology, Inc.* Web. 25 Nov. 2013. <http://www.bdti.com/MyBDTI/pubs/choose_2000.pdf>.

7. "Choosing FPGA or DSP for Your Application." *FPGA or DSP*. Hunt Engineering, n.d. Web. 23 Nov. 2013. <http://www.hunteng.co.uk/info/fpga-or-dsp.htm>.

8. "Comparison of 3 DTTB Systems." *Comparison of 3 DTTB Systems*. Digital Broadcasting Experts Group, n.d. Web. 24 Nov. 2013. <http://www.dibeg.org/techp/3comp/3comp.html>.

9. "Component Video Definition." *PCmag*. PCmag. Web. 1 Dec 2013. <http://www.pcmag.com/encyclopedia/term/40115/component-video>.

10. "Coprocessor Interface." *ARM Information Center*. ARM. Web. 1 Dec 2013. <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0211k/Cacicebd.html>.

11. "CURRENT-LIMITED, POWER-DISTRIBUTION SWITCHES" Texas Instruments. Web. 27 Apr. 2014. <http://www.ti.com/lit/ds/symlink/tps2052b.pdf>.

12. "Digital Video Broadcasting - Cable (DVB-C)." *Techopedias*. N.p., n.d. Web. 24 Nov. 2013. <http://www.techopedia.com/definition/25172/digital-video-broadcasting-cable-dvb-c>.

13. "DIRECTPATH™ STEREO LINE DRIVER, ADJUSTABLE GAIN" Texas Instruments. Web. 27 Apr. 2014. <http://www.ti.com/lit/ds/symlink/drv601.pdf>.

14. "DM3730, DM3725 Digital Media Processors" Texas Instruments. Web. 27 Apr. 2014. <http://www.ti.com/lit/ds/symlink/dm3730.pdf>.

15. "DS6000 Customer Information Center." *DS6000 Customer Information Center*. IBM, n.d. Web. 01 Dec. 2013. <http://publib.boulder.ibm.com/infocenter/dsichelp/ds6000ic/index.jsp?topic=/com.ibm.storage.smric.help.doc/f2d_smresd_1td5nf.html>.

16. "DTMB Technology." *DTMB Technology*. Rohde & Schwarz, n.d. Web. 24 Nov. 2013. <http://www.rohde-schwarz.com/en/technologies/terrestrial-broadcast/dtmb/dtmb-technology/dtmb_technology_55770.html>.

17. "EP5368QI" ENPIRION. Web. 27 Apr. 2014. <http://www.mouser.com/catalog/specsheets/EP5368QI_03260F.pdf>.

18. "FPGA vs. ASIC." *FPGA vs. ASIC*. Xilinx, n.d. Web. 23 Nov. 2013. <http://www.xilinx.com/fpga/asic.htm>.

19. "FPGAs vs. DSPs: A Look at the Unanswered Questions." *EETimes*. Berkeley Design Technology, Inc., n.d. Web. 23 Nov. 2013. <http://www.eetimes.com/document.asp?doc_id=1275357>.

20. "FT232R USB UART I.C." FTDI Ltd. Web. 27 Apr. 2014. <www.ftdichip.com/Documents/**DataSheets**/ICs/DS_FT232R.pdf>.

21. "Fundamentals of Digital Image Processing." Hong Kong Polytechnic University, n.d. Web. 28 Nov. 2013. <http://www.eie.polyu.edu.hk/~enyhchan/imagef.pdf>.

22. "HDMI Companion Chip with I 2C Level Shifting Buffer, 12 Channel ESD, and Current-Limit Load Switch."*Texas Instuments*. Texas Insturments, n.d. Web. 1 Dec 2013.

23. "High-Performance 16-bit Non-PCI 10/100 Ethernet Controller with Variable Voltage I/O " SMSC. Web. 27 Apr. 2014. <http://ww1.microchip.com/downloads/en/DeviceDoc/9221.pdf>.

24. "Hot Swap." *What Is ?* N.p., n.d. Web. 28 Nov. 2013. <http://whatis.techtarget.com/definition/hot-swap>.

25. "IDE OverView|QtCreator." *Qt-Project*. Qt. Web. 1 Dec 2013. <http://qt-project.org/doc/qtcreator-2.8/creator-overview.html>.

26. "Integrated Development Environment." *Webopedia*. N.p.. Web. 1 Dec 2013. <http://www.webopedia.com/TERM/I/integrated_development_environment.html>.

27. "Introduction to FPGA Technology: Top 5 Benefits." *National Instruments*. National Instruments, 16 Apr. 2012. Web. 23 Nov. 2013. <http://www.ni.com/white-paper/6984/en/>.

28. "LM2678 SIMPLE SWITCHER® High Efficiency 5A Step-Down Voltage Regulator." *Texas Instuments*. Texas Insturments, n.d. Web. 1 Dec 2013. <http://www.ti.com.cn/cn/lit/ds/symlink/lm2678.pdf>.

29. "Lowest Component Count, Energy-Effi Cient Off-Line Switcher IC." *LinkSwitch-TN Family*. Power Integrations, n.d. Web. 1 Dec. 2013. <http://www.wvshare.com/datasheet/POWER_PDF/LNK304.PDF>.

30. "MIC5247" MICREL. Web. 27 Apr. 2014. <http://www.micrel.com/_PDF/mic5247.pdf>.

31. "MPEG-2 Video Encoding (H.262)." *Sustainability of Digital Formats Planning for Library of Congress Collections*. Library of Congress, 1 Dec. 2012. Web. 29 Nov. 2013. <http://www.digitalpreservation.gov/formats/fdd/fdd000028.shtml>.

32. "Native Command Queuing." *Welcome to SATA-IO*. Serial ATA, n.d. Web. 28 Nov. 2013. <https://www.sata-io.org/native-command-queuing>.

33. "OMAP Applications Processors." *Texas Instruments*. Texas Instruments. Web. 1 Dec 2013. <http://www.ti.com/lsds/ti/omap-applications-processors/overview.page>.

34. "PAL & NTSC & SECAM." *Filmbug Birthdays RSS*. Filmbug, n.d. Web. 24 Nov. 2013. <http://www.filmbug.com/dictionary/pal-ntsc.php>.

35. "PJ1-022-SMT" CUI, Inc. Web. 27 Apr. 2014. <http://www.cui.com/product/resource/pj1-022-smt.pdf>.

36. "Quadrature Amplitude Modulation (QAM)." *Quadrature Amplitude Modulation (QAM)*. National Instruments, 01 Feb. 2012. Web. 22 Nov. 2013. <http://www.ni.com/white-paper/3896/en/>.

37. "RAID (redundant Array of Independent Disks)." *What Is RAID?* TechTarget, n.d. Web. 26 Nov. 2013. <http://searchstorage.techtarget.com/definition/RAID>.

38. "STMicroelectronics Kyro II 64MB." *RSS*. N.p., n.d. Web. 28 Apr. 2014.

39. "TFP410" Texas Instruments. Web. 27 Apr. 2014. <http://www.ti.com/lit/ds/symlink/tfp410.pdf>.

40. "TMS320C6000 DSP Multichannel Audio Serial Port (McASP)." *Texas Instruments*(2008): n. pag. *Texas Instruments*. Web. 27 Nov. 2013.

41. "Transformerless switch mode power supply circuit."*Circuits Today*. N.p., 28 Jun 2011. Web. 1 Dec 2013. <http://www.circuitstoday.com/transformerless-switch-mode-power-supply-circuit>.

42. "ULTRA-LOW POWER, LOW INPUT VOLTAGE, CURRENT-LIMITED LOAD SWITCH WITH SHUT-OFF, AUTO-RESTART, AND OVER-CURRENT CONDITION TIME-OUT." *Texas Instuments*. Texas Insturments, n.d. Web. 1 Dec 2013. <http://www.ti.com/lit/ds/symlink/tps22946.pdf>.

43. "Universal Serial Bus Specification." *Powered USB*. Compaq, Hewlett Packard, Intel, Lucent, Microsoft, NEC, Phillips, Apr. 2000. Web. 1 Dec. 2013. <http://www.poweredusb.org/pdf/usb20.pdf>.

44. "USB 2.0 Hi-Speed Hub Controller " SMSC. Web. 27 Apr. 2014. < http://media.digikey.com/pdf/Data%20Sheets/SMSC/USB2512,13,14,17(A,B).pdf>.

45. "USB." *What Is (Universal Serial Bus)* Computer Hope, n.d. Web. 26 Nov. 2013. <http://www.computerhope.com/jargon/u/usb.htm>.

46. "Watch, Pause and Record TV on Your Windows PC or Laptop." *Hauppauge Computer Works : WinTV-HVR-950Q Product Description*. Hauppauge, n.d. Web. 28 Apr. 2014.

47. "What Is DVB-T?" *WiseGEEK*. N.p., n.d. Web. 24 Nov. 2013. <http://www.wisegeek.org/what-is-dvb-t.htm>.

48. "Why ReplayTV." *ReplayTV*. N.p.. Web. 30 Nov 2013. <http://www.digitalnetworksna.com/about/replaytv/>.

49. . *Arch Linux*. Arch Linux. Web. 1 Dec 2013. <https://wiki.archlinux.org/index.php/Arch_Linu&xgt>.

50. 26. "Ultra High Frequency." *WolframAlpha*. WolframAlpha. Web. 1 Dec 2013. <http://www.wolframalpha.com/input/?i=ultra high frequency>.

51. Afra, Bamdad, and Amit Kapadiya. "Making design choices between DSP and FPGA." *EETimes*. N.p., 13 May 2008. Web. 23 Nov. 2013. <http://www.eetimes.com/document.asp?doc_id=1271631>.

52. Angoletta, Maria-Elena. "Digital Signal Processor Fundamentals and System Design." *CERN Document Server*. CERN, 9 June 2007. Web. 26 Nov. 2013. <http://cds.cern.ch/record/1100536?ln=en>.

53. ArchLinux, . "About|Arch Linux Arm." *Arch Linux*. Arch Linux. Web. 1 Dec 2013. <http://archlinuxarm.org/about>.

54. ARM, . "ARM Information Center." . ARM. Web. 1 Dec 2013. <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.home/index.html>.

55. ARM, . "ARM Processor Architecture." *ARM Information Center*. N.p.. Web. 1 Dec 2013. <http://www.arm.com/products/processors/instruction-set-architectures/index.php>.

56. Brain, Marshall, William Harris, and Robert Lamb. "How Electricity Works."*HowStuffWorks*. N.p., n.d. Web. 01 Dec. 2013. <http://science.howstuffworks.com/electricity9.htm>.

57. Brain, Marshall. "How Television Works" 26 November 2006. HowStuffWorks.com. <http://electronics.howstuffworks.com/tv.htm> 23 November 2013.

58. Brain, Marshall. "How USB Ports Work." *How USB Ports Work*. How Stuff Works, n.d. Web. 26 Nov. 2013. <http://computer.howstuffworks.com/usb3.htm>.

59. Brain, Marshall. "Inside a TV Remote Control." *How Stuff Works?*. N.p., n.d. Web. 1 Dec 2013. <http://electronics.howstuffworks.com/inside-rc1.htm>.

60. Brenner, Neal. "Ultra-Low Power TV IR Remote Control Transmitter." *Texas Instuments*. Texas Insturments, n.d. Web. 1 Dec

2013. <http://www.ti.com/lit/an/slla175/slla175.pdf>.

61. BroadCom, . "High Definition 1080p Embedded Multimedia Applications Processor - BCM2835." *Broadcom*. N.p.. Web. 1 Dec 2013. <http://www.broadcom.com/products/BCM2835>.

62. Day, Michael. "Understanding Low Drop Out (LDO) Regulators ." *Texas Instuments*. Texas Insturments. Web. 1 Dec 2013. <http://focus.ti.com/download/trng/docs/seminar/Topic 9 - Understanding LDO dropout.pdf>.

63. Dimitrova, N, S Jeannin, J Nesvadba, T McGee, L Agnihotri, and G Mekenkamp. "Real time commercial detection using MPEG features." . N.p.. Web. 1 Dec 2013. <http://jan.nesvadba.info/cv/2002 IPMU RealTime Commercial Detection Using MPEG Features.pdf>.

64. Dubash, Manek. "MLC vs SLC: Which flash SSD is right for you?." *ComputerWeekly.com*. ComputerWeekly.com. Web. 1 Dec 2013. <http://www.computerweekly.com/feature/MLC-vs-SLC-Which-flash-SSD-is-right-for-you>.

65. Duygulu, Pınar, Ming-yu Chen, and Alexander Hauptmann. "Comparison and Combination of Two Novel Commercial Detection Methods." . N.p.. Web. 1 Dec 2013. <http://lastlaugh.inf.cs.cmu.edu/alex/duygulu_ICME04.pdf>.

66. Ekker, Neal, Tom Coughlin, and Jim Handy. "Solid State Storage 101." . SNIA, n.d. Web. 1 Dec 2013. <https://members.snia.org/apps/group_public/download.php/35796/SSSI Wht Paper Final.pdf>.

67. Erik. "A Short Guide to Comskip." *Tuning Comskip*. N.p., n.d. Web. 01 Dec. 2013. <http://www.kaashoek.com/files/manual.htm>.

68. Gosselin, Brian. "Ditch the NTC Thermistor: Use an Analog Temp Sensor." *TI E2E Community*. Texas Instruments, 31 May 2013. Web. 28 Nov. 2013. <http://e2e.ti.com/blogs_/b/analogwire/archive/2013/05/31/ditch-the-ntc-thermistor-use-an-analog-temp-sensor.aspx>.

69. Igarta, Michael. *A Study of MPEG-2 and H.264 Video Coding*. Thesis. Purdue University, 2004. N.p.: n.p., n.d. Print.

70. Jacky . "5V buck regulator using LM2678." *Circuits Today*. N.p., 15 Jan 2010. Web. 1 Dec 2013. <http://www.circuitstoday.com/5v-buck-regulator-using-lm2678>.

71. Jung, Walt and Adolfo Garcia, "Op Amps in Line-Driver and Receiver Circuits, Part 2," *Analog Dialogue*, 27-1, 1993, pp. 14 - 17. Op Amps in Line-Driver and Receiver Circuits, Pt 2

72. Kalinsky, David, and Roee Kalinsky. "Introduction to I2C." *Embedded*. N.p., 31 July 2001. Web. 26 Nov. 2013. <http://www.embedded.com/electronics-blogs/beginner-s-corner/4023816/Introduction-to-I2C>.

73. Kayne, R., and L. S. Wynn. "What Is Ethernet?" *WiseGeek*. N.p., 29 Oct. 2013. Web. 26 Nov. 2013. <http://www.wisegeek.com/what-is-ethernet.htm>.

74. Kerekes, Zsolt, ed. "SSD Jargon Explained."*StorageSearch*. StorageSearch.com. Web. 1 Dec 2013. <http://www.storagesearch.com/ssd-jargon.html>.

75. Kester, Walt, and James Bryant. "Grounding in Mixed Signal Systems." *Hardware Design Techniques* (n.d.): n. pag. *Analog Devices*. Web. 27 Nov. 2013. <http://www.analog.com/static/imported-files/analog_dialogue/5467026043687049331665676350Grounding.pdf>.

76. Lambert, Bryan. "Solid Stat Hard Drives." *Geeks.com*. Geek, 11 Jan 2009. Web. 1 Dec 2013. <http://www.geeks.com/techtips/2009/techtips-11JAN09.htm>.

77. Lazaridis, Giorgos. "The RC5 Protocol specifications." *PCB Heaven*. N.p., 30 Jul 2012. Web. 1 Dec 2013. <http://www.pcbheaven.com/userpages/The_Philips_RC5_Protocol/>.

78. Mischel, Jim. "Re-Thinking User Interface Design for the TV Platform." *Smart Bear*. N.p., 12 May 2011. Web. 29 Nov 2013. <http://blog.smartbear.com/how-to/re-thinking-user-interface-design-for-the-tv-platform/>.

79. N.p.. Web. 29 Nov 2013. <http://www.kaashoek.com/files/manual.htm>.

80. Nist, Ken. "HDTV Primer." *HDTV Primer*. N.p., 10 Dec. 2009. Web. 24 Nov. 2013. <http://www.hdtvprimer.com/>.

81. Palermo, Samuel Michael. *A Multi-band Phase-locked Loop Frequency Synthesizer*. Thesis. Texas A & M University, 1999. N.p.: n.p., n.d. Print.

82. Patel, Raoji, and Glenn Fritz. "Switching Power Supply Design Review - 60 Watt Flyback Regulator." . Texas Insturments, n.d. Web. 1 Dec 2013. <http://www.ti.com/lit/ml/slup072/slup072.pdf>.

83. Pidgeon, Nick. "How Ethernet Works." *HowStuffWorks*. N.p., n.d. Web. 26 Nov. 2013. <http://computer.howstuffworks.com/ethernet4.htm>.

84. Pirillo, Chris. "What is the Difference Between Comosite and Component Video?." *chris.pirillo.com*. N.p.. Web. 1 Dec 2013. <http://chris.pirillo.com/what-is-the-difference-between-composite-and-component/>.

85. Poynton, Charles. *Digital Video and HDTV*. San Francisco: Morgan Kaufmann Publishers, 2003. Print.

86. Qt, . "QtCreator." *Qt-Project*. Qt-Project. Web. 1 Dec 2013. <http://qt-project.org/wiki/Category:Tools::QtCreator>.

87. Rai, Seema. "Direct Memory Access." (): n. pag. Print.

88. Roggen, Daniel. "Qt Media Encoding Library." *QtMEL*. N.p.. Web. 1 Dec 2013. <http://kibsoft.ru/>.

89. Satterwhite, Brandon, and Oge Marques. *IEE Potentials*. IEE, n.d. Web. 29 Nov 2013. <http://www.cse.unr.edu/~bebis/CS474/StudentPaperPresentations/DetectionTVCommercials.pdf>.

90. Schweber, Bill. "TV Tuner IC Captures Analog, Digital Standards, Features LNA and Tracking Filters for Superior Performance." *TV Tuner IC Captures Analog, Digital Standards, Features LNA and Tracking Filters for Superior Performance*. EETimes, 29 June 2009. Web. 24 Nov. 2013. <http://www.eetimes.com/document.asp?doc_id=1312442>.

91. Skolnick, David, and Noam Levine. "Why Use DSP?" *Analog Devices: Analog Dialogue: Digital Signal Processing 101 An Introductory Course in DSP System Design: Part 1*. Analog Devices, n.d. Web. 23 Nov. 2013. <http://www.analog.com/library/analogDialogue/archives/31-1/DSP.html>.

92. Strickland, Jonathan, and James Bickers. "How DVR Works." *HowStuffWorks*. N.p., n.d. Web. 29 Nov. 2013. <http://electronics.howstuffworks.com/dvr.htm>.

93. Tanenbaum, . Computer Networks 4th Edition. Print.

94. Tatum, Malcolm, and Bronwyn Harris. "What Is a Line Driver." *WiseGeek*. N.p., 21 Oct. 2013. Web. 24 Nov. 2013. <http://www.wisegeek.com/what-is-a-line-driver.htm>.

95. Texas Instruments, . "OMAP Applications Processors."*Texas Instruments*. N.p.. Web. 1 Dec 2013. <http://www.ti.com/lsds/ti/omap-applications-processors/products.page

96. Texas Instruments, . "OMAP™ 4 Processors." *OMAP Applications Processor*. Texas Instruments. Web. 1 Dec 2013. <http://www.ti.com/lsds/ti/omap-applications-processors/omap-4-processors-products.page?paramCriteria=no>.

97. Torres, Gabriel. "Everything You Need To Know About DDR, DDR2 and DDR3 Memories | Hardware Secrets." *Everything You Need To Know About DDR, DDR2 and DDR3 Memories | Hardware Secrets*. Hardware Secrets, 27 Aug. 2009. Web. 28 Nov. 2013. <http://www.hardwaresecrets.com/article/Everything-You-Need-To-Know-About-DDR-DDR2-and-DDR3-Memories/167/7>.

98. Torres, Gabriel. "Everything You Need to Know About the SPDIF Connection | Hardware Secrets." *Hardware Secrets*. N.p., 18 July 2011. Web. 30 Nov. 2013. <http://www.hardwaresecrets.com/printpage/Everything-You-Need-to-Know-About-the-SPDIF-Connection/82>.

99. Torres, Gabriel. "How Analog-to-Digital Converter (ADC) Works." *Hardware Secrets*. N.p., 21 Apr. 2006. Web. 30 Nov. 2013. <http://www.hardwaresecrets.com/article/How-Analog-to-Digital-Converter-ADC-Works/317/1>.

100. TV Fool, . "TV Signal Analysis Result." *TV Fool*. TV Fool. Web. 1 Dec 2013. <http://www.tvfool.com/?option=com_wrapper&Itemid=29&q=id=46ae26c69dd123>.

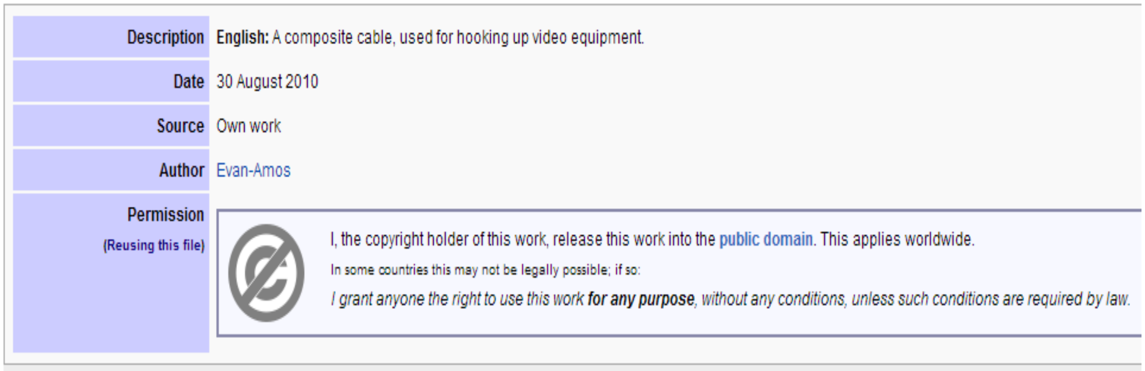
101. Wain, Richard. *An Overview of FPGAs and FPGA Programming: Initial Experiences at Daresbury*. Warrington: Council for the Central Laboratory of the Research Councils, 2006. Print.

102. Wang, Yiwei, and John F. Doherty. "Moving Object Tracking in Video." (n.d.): n. pag.*National Institute of Standards and Technology*. Web. 29 Nov. 2013. <http://www.antd.nist.gov/pubs/aipr00.pdf>.

103. wiseGeek, . "What is Composite Video." *wiseGeek*. wiseGeek. Web. 1 Dec 2013. <http://www.wisegeek.com/what-is-composite-video.htm>.

Appendix B

Permissions



Description	**English:** A composite cable, used for hooking up video equipment.
Date	30 August 2010
Source	Own work
Author	Evan-Amos
Permission (Reusing this file)	I, the copyright holder of this work, release this work into the public domain. This applies worldwide. In some countries this may not be legally possible; if so: *I grant anyone the right to use this work* ***for any purpose****, without any conditions, unless such conditions are required by law.*

All other figures were original pictures or redrawn using Microsoft Visio and Powerpoint.