# AUDIO CONTROLLED PROCESSOR

by

Brian Victor Michon

Submitted in Partial Fulfillment

of the Requirements for the

Degree of Bachelor of Science

at the

Massachusetts Institute of Technology

May, 1982

Signature of Author..........................................
Department of Electrical Engineering
and Computer Science, May, 1982

Certified by..........................................
Thesis Supervisor

Accepted by..........................................
Chairman, Departmental committee on
Theses

# AUDIO CONTROLLED PROCESSOR

by

Brian Victor Michon

Submitted in Partial Fulfillment of the Requirements for the degree of Bachelor of Science at the Massachusetts Institute of Technology, May, 1982.

## Abstract

The purpose of this thesis was to design, construct and apply a device capable of modifying a video images color by keying on the frequency of an audio source. An Audio Controlled Processor (ACP) is an analog and digital circuit designed to be placed in series between an RGB monitor and a RGB driver. The ACP provides for eight variable frequency keying regions, that if a frequency is detected in that range the intensity of the red, green or blue color vector can be increased or decreased by a programable amount. A prototype ACP was constructed and tested. The device met all functional specification but produced an undesirable flashing effect inherent to its initial design.

Thesis Supervisor: Benjamin Bergery
Title: Faculty, Film/Video Department
Massachusetts Institute of Technology

# Table of Contents

List of Figures

## The Problem and the Interest

The video medium has been growing steadily for the last 30 years. Its growth rate has increased further with the advent of cable television. Growing along with the video medium is the assorted technology of video effect generators. Devices used by broadcasters, video engineers and artists to modify video signals in an interesting and aesthetic manner are in growing demand.

The types of video processing equipment are numerous. There exist proc-amps, special effects generators, video mixers, keyers and others. But out of all this equipment I have seen no device that modifies video using audio as a control input.

Using audio as a control input is extremely common in other applications. There exists sound activated switches, speech recognition systems and sonar to name a few. Perhaps the closest application of an audio controlled device to a video application is a color organ, a device that turns color lights on in the presence of a specified audio frequency.

The question is whether an audio controlled device could be created that would modify video in an interesting and aesthetic manner. Also, if could be created, what particular applications would it be useful for.

# History of Project

The idea of using audio to manipulate a video image first came to me last spring. A student in Introductory Movie Making Workshop at MIT was making a presentation on black and white video digitization. During his demonstration he showed the effects of dropping bits ( asserting logical 1 or 0 ) from the digitized signal. Intrigued by the effect this had on the image I pondered the idea of methodically selecting which bits to drop. When in the demonstration I saw the effect of dropping the most significant bit of a dance tape, I realized that selecting dropout bits on the basis of the video audio source would have a highly interesting effect.

My initial idea was to build an audio digitizer and using combinatorial logic have the digitized audio frequency select which bits to drop from the digitized video signal. Thinking that this was appropriate thesis material I contacted Benjamin Bergery of the MIT's Film/Video department. We found the interest mutual and Benjamin agreed to be my thesis supervisor.

Both Benjamin and I desired that we use color video. This presented a fundamental problem, the availability of a color digitizer, a rather large and expensive device. I contacted Ron McNiel of MIT's Visual Language Workshop to see if I could use their Grinnel. I discovered that to get at the digitized video signal would require the cutting of bus lines in the Grinnel.

I was apprehensive of this idea for I wanted a more modular device that could be applied to almost any system. Ron suggested that I abandon the idea of using a video digitizer and do my processing on the analog red, green and blue (RGB) video signals.

With this approach, instead of dropping bits of the digitized video signal, processing would be done by increasing or decreasing the intensity of the analog signal. The effect, although different, would be a highly interesting color manipulation of the video. The greatest advantage to this approach is that the device could be run off any RGB driver.

## Available Literature

The information needed to construct the ACP was abundant. I found numerous texts that were virtually cookbooks for the different circuits needed. I also found a large number of television books that explained the video signal, although my lecture notes from Introductory Movie Making Workshop at MIT were quite sufficient.

Literature specifically on a device like the ACP, however, was not to be found. I searched the libraries at MIT, BU, Simmons, Wellesly and Boston Public. I considered having a literature search done but didn't because it requires specific search words which in this case would have turned out to be expensive guesses. Searching for the string "audio controlled processor" would have undoubtedly turned up nothing for there are no standard names in this area.

Literature on video processors was accessible. In particular I found one MIT thesis that, while offering little relevant technical information, was a useful overview to the project. All relevant references are listed in the bibliography.

## RGB Processing

Color video is based on mixing three primary color vectors: red, green and blue. Color video is an additive process. Asserting equal intensities of red and blue to a color monitor produces magenta. Mixing equal intensities of red, green and blue produces white.

Different saturations of the color's hue are obtained by mixing more or less white. Figure 1 is a color triangle or CIE chromaticity diagram. The vertices represent the primary colors red, green and blue. Moving from one vertex to another along the perimeter all the hues in nature can be produced. Different saturations are made by moving toward the center of the triangle.

RGB color systems apply this additive mixing principle to generate video. Basically a red, green and blue color picture are transmitted simultaneously. These three primary video signals signals are then superimposed to obtain a color composite picture.

By increasing or decreasing the intensity of an RGB color vector you respectively move towards or away from the corresponding vertex on the color triangle. Example, on figure 1 if you are at point A(yellow) and increase the blue you move towards the white region point B. And if you are at point D(magenta) and decrease the blue the result is red or point C.

This is the exact process in which the ACP will modify the video's color. When a key frequency is detected the ACP will either increase or decrease a color vector causing a movement in the color triangle of the video image's color.

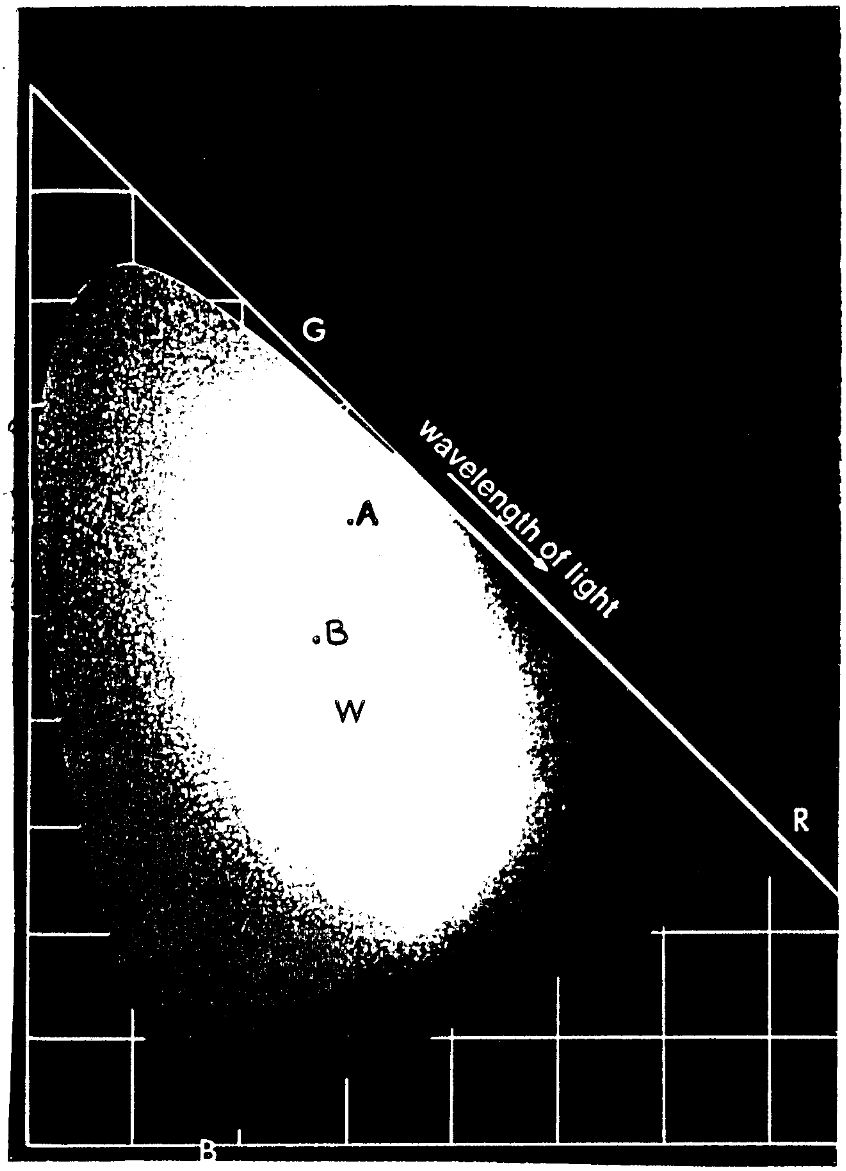


Figure 1. CIE Chromaticity Graph

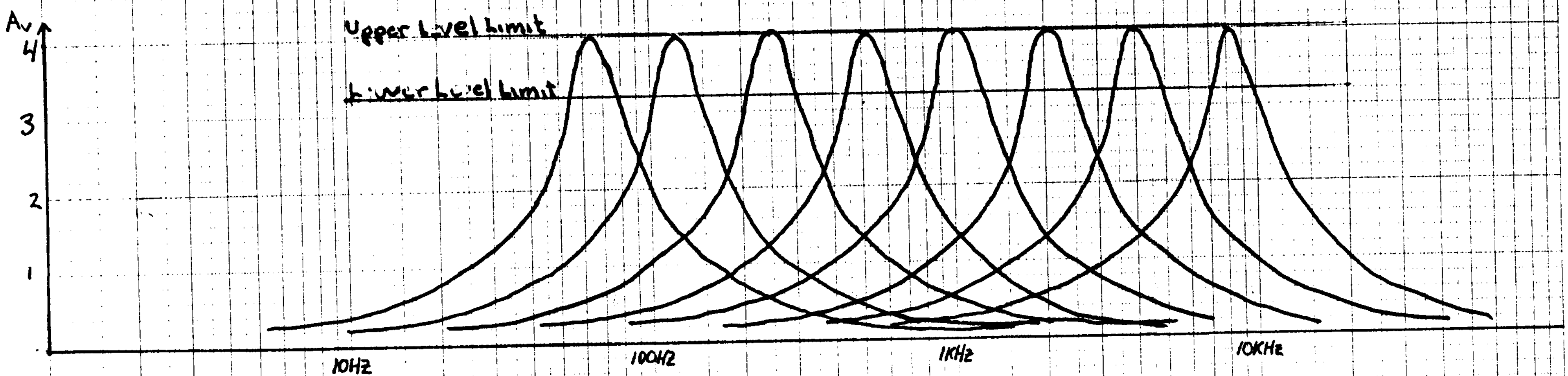


Figure 2 Filter Frequency Response

## Formal Specifications

The function of the Audio Controlled Processor (ACP) is to increase or decrease the intensity of the red, green and blue (RGB) color vectors of a video image if an audio frequency is detected in a keying range.

The ACP requires six inputs and has three outputs. The first three are the red, green and blue video signal color vectors. The fourth input is an audio source. The fifth and sixth inputs are video sync signals, namely composite sync and vertical drive. Outputs one through three are the modified color vectors.

Functional specifications: There are eight audio frequency keying regions, with variable ranges of 0 - 25% of center frequencies 64Hz, 125Hz, 250Hz, 500Hz, 1KHz, 2KHz, 4KHz and 8KHz. Each of these keying regions may be used as keys to modify the intensity of the red, green or blue video color vectors. The effect a detected key frequency has on the color is programable in three different ways. One, the color vector can be increased or decreased in intensity. Two, the amount of increase or decrease is variable from zero to full video intensity. And three, the ACP is also responsive to frequency variations detected within a particular keying window.

Input specifications: The red, green and blue video signals of 0.0 to 0.7 volts (terminated at 75 ohms) with blanking optional. The audio source is to be 2 volts peak to peak(terminated at 10M ohm). The video sync signals should be standard, the only restriction being that they be asserted negative(terminated at 10M ohm).

Output specifications: The modified red, green and blue output color vectors should be within a 0.0 to 0.7 volt range when terminated at 75 ohms. These outputs must also be clamped at 0.0 volts during composite sync.

## Circuit Implementation

### Outline

The circuit can be broken down into four sections. The first section is the audio filters. Eight active gain band pass filters are used to break the audio input into designated frequency ranges.

The second section is the frequency to voltage converter. Once the audio input is filtered it is passed through level detectors that produce clock pulses for each key cycle. The number of pulses in a video frame is then multiplied by a settable number between 0 and 256. The result is then fed to a digital to analog converter to produce a voltage in a range between 0.0 and 0.7 volts.

The third section is the color mixers. Here the D to A's voltage is added or subtracted from the RGB color vector. The result is a new intensity of the primary color vector if a keying frequency is detected.

The final section cleans up the modified video signals. 0.0 volts is restored to the vector during composite sync. The output voltage of the vector is clipped to be in a 0.0 to 0.7 voltage window. The final function is to drive the video output at a maximum current of 10mA.

## Filters

The design of active band pass filters is an art in itself. An excellent way to determine a configuration is to search the literature. I found an appropriate design in the National Semiconductor Audio Handbook. The design was particularly convenient since the values worked out to be easily obtainable resistors and capacitors, and gain and a high Q are present.

Figure 2 is a graph of the frequency response of the filters constructed. Indicated on the figure are the limits of the level detectors(note: the audio input is two volts peak to peak thus the graph represents relative gain as well as absolute levels.) the bands are quite disjointed and there is a wide range of frequencies available for triggering.

## Frequency to Voltage

This section is basically a programable digital to analog converter. The filtered audio is passed to a level detector. If the amplitude of the filtered audio is higher than the level set at the detector a pulse is generated. Figure 4 is a schematic and functional diagram of the level detector used. The TTL pulse is generated by a unipolar limiter, see figure 5.

The next phase of the frequency to voltage converter is a digital multiplier. The number of key pulses in one video frame (1/30 second) are multiplied by a programable number. This result is then latched to a D to A with an adjustable maximum output from 0.0 to 0.7 volts and a range from 0.0

to maximum. There are many ways to build a multiplier and just as many reasons for doing it. I chose my particular design because I had most of the components for this implementation. The basic idea is an eight bit adder for each color; one of the inputs is the programable eight bit multiplier, the other input is a latched output of the same adder. Each key pulse latches the new output back to the input of the adder. The result is the number of key pulse times the programable multiplier. To stop overflow the latch is disabled when the adder is near overflow.

At the start of a new video frame this result is latched again, this time to a D to A. Also the "multiplier" is cleared and starts counting again. This output on the D to A is used to create the voltage offset that acts to increase or decrease the color vectors.

An example of how this works: Let's choose the 1KHz filter set at its highest level (ie. narrowest band width) to key an increase in the red vector. We set the D to A range to respond bijectively to the eight bit input with an output range of 0.0 to 0.7 volts (full intensity). If we set the multiplier to its full position (256) and at least one key frequency of 1KHz is detected the multiplier will output full at 256 and the D to A will be at 0.7 volts for the next frame. If we change the multiplier to one, the number of 1KHz cycles detected in a frame(1/30 sec.) will be the input to the D to A. BUT... since we are only sampling for 1/30 of a second only 33 1KHz cycles can be detected. Thus the highest input to the D to A can be 33 and the highest voltage output is 33/256 X 0.7 = 0.09 volts. To have a more pronounced effect on the number of 1KHz keys in a frame the multiplier should be set at 256/33 = 7(approx.)

## Color Mixers

The color mixers are simply electrical voltage adders and subtractors. The red, green and blue color vectors are terminated at 75 ohms in the ACP to bring them to standard video levels. The other functional inputs are from the digital to analog converters of the frequency to voltage section. These inputs are also at standard video levels. Since no amplification is needed on the video signal the adders and subtractors operate at unity gain. See figures 6 and 7.

## Video Driver

The video driver section simply cleans up the modified red, green and blue color vectors. First the signal is passed through a voltage window with limits of 0.0 volts and 0.7 volts. This is simply a zener diode limiter circuit. After the voltage is window limited, sync levels are restored to the vectors by grounding the signal with an open-collector nand gate during composite sync. The final stage is the driver itself. This is just a low output impedance driver capable of outputting 10 milliamps of current. These designs are fairly simple, refer to figure 9 for actual implementation details.

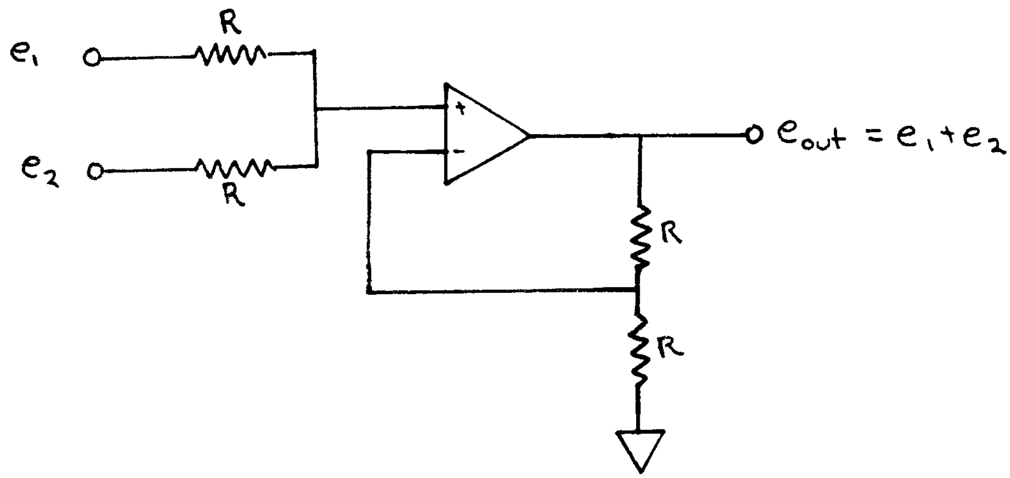


Figure 6 Voltage Adder

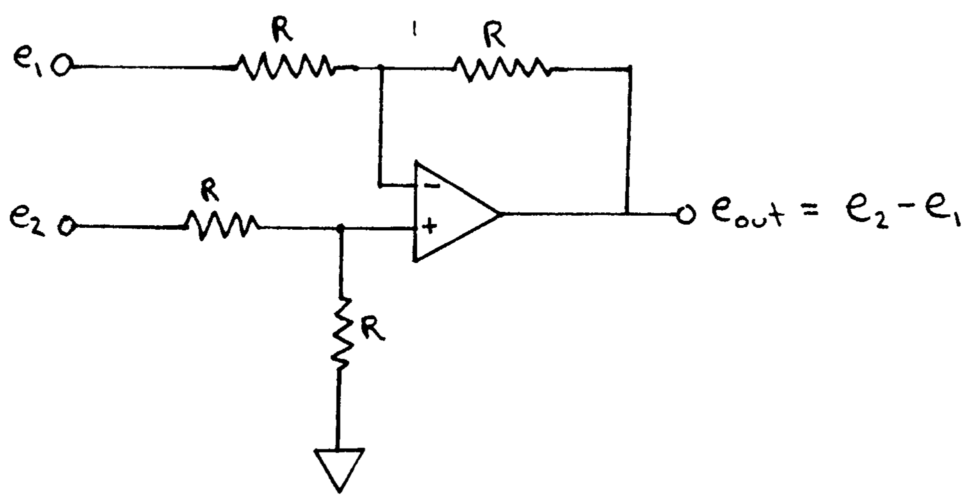


Figure 7 Voltage Subtractor

## A Word on Components

When working with video signals it is necessary to use fairly fast components capable of handling the 5MHz video bandwidth. I found that the LM318 operational amplifier to be a good choice. It has a typical bandwidth of 15MHz and a slew rate of 50V/usec. A single .01 Mf capacitor reduces settling time to under a microsecond. Furthermore the LM318 is internally compensated for unity gain and cost under a dollar for small quantities.

Components other than video processing op amps did not have to meet as high specifications. I chose these components by what was available and what was most economical.

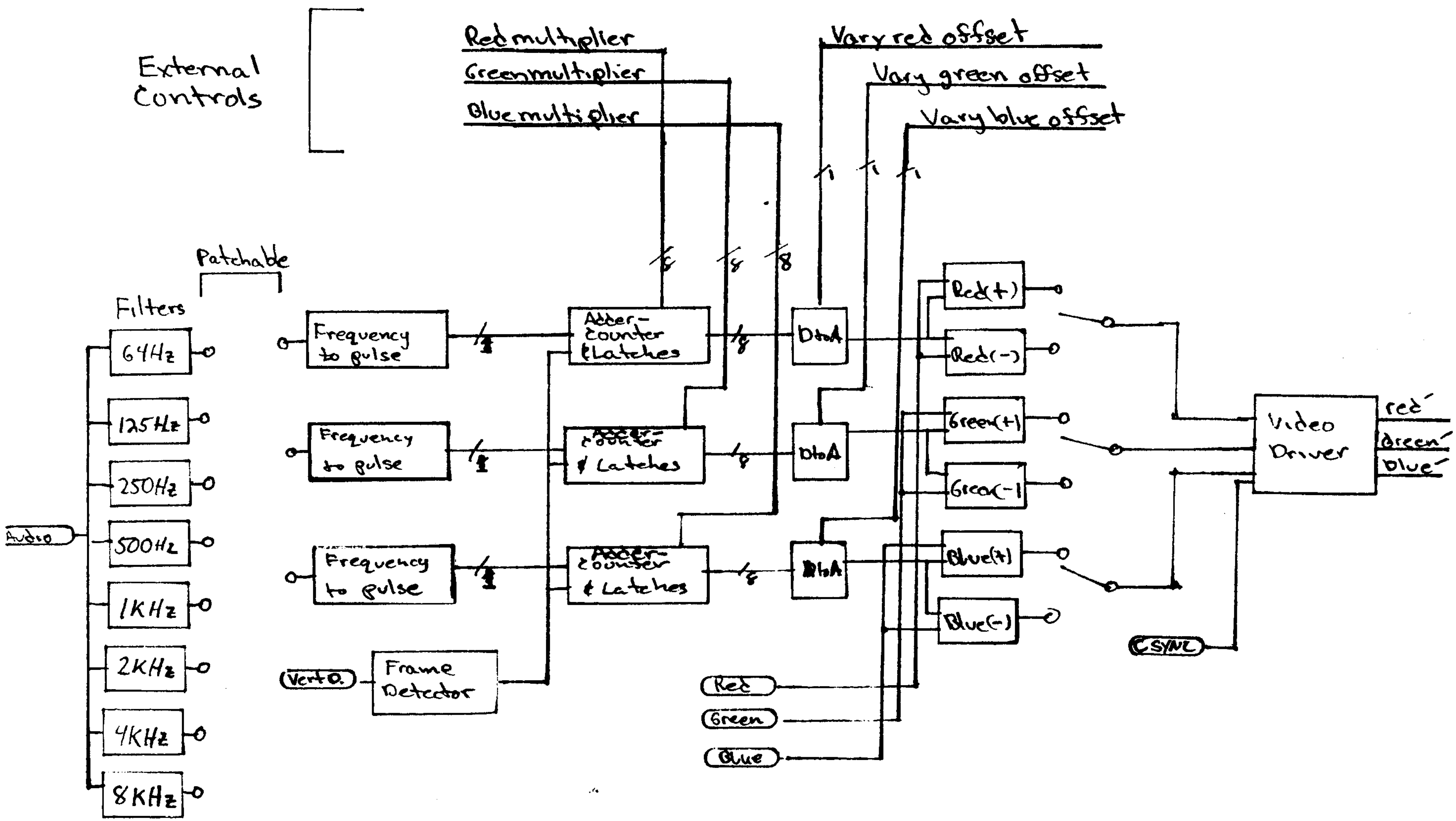


Figure 8 Block Diagram

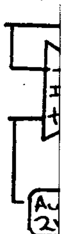

Figure 9 Schematic

## Actual Effects

Like so many things in life, the actual effect of the ACP was unexpected. From a functional point of view the device met almost every specification. From an aesthetic point of view the ACP produced a flashing display that was difficult to look at.

The flashing effect can be described as a sudden overall color increase of the entire display at about a 3Hz rate. In other words, if the video image was a red apple in a grey room the apple would flash bright red but the room would also flash pink. Since this rate is perceptibly fast, the flashing of the screen is difficult to look at.

The result of increasing the intensity of a frame's color vector was interesting but not pleasant. For example setting the ACP to respond in full to any frequency in a keying range(multiplier of 256), producing a modifying intensity of one half full video(0.35 volts) level and setting the ACP to add to the red vector causes the light red objects to become bright red but also causes the whole screen to flash red. This screen flashing is difficult to look at due to its strobing affect.

No combination of controls produced an image that was interesting but didn't flash. It didn't matter what frequency was triggering the color the screen still flashed at approximately 3Hz. Adjusting the multiplier to respond to the number of keys in a frame not only didn't remove the flashing but could not be perceived at all. Varying the intensity of the modification had the most pronounced effect, but still could not remove flashing while maintaining interesting color shift. Switching from addition of the modification to subtraction caused the complimentary colors to shift, but alas the flashing remained.

The ACP could not produce an image that didn't flash for the cause lay in its circuitry. Color modification is done by addition and subtraction of the color evenly over the whole video frame. This is the direct cause of the screen flashing. The problem is that the ACP introduces a constant offset to the color. What should have been is to introduce an offset only to existing colors. This will eliminate frame flashing and only produce desirable object flashing. The implementation of this is straight forward: simply replace the voltage adders and subtractors with other functions that use the existing color as a parameter, ie. multipliers or power functions.

Another approach to solving the flashing problem involves reducing the rate of modification to intervals where screen flashing isn't annoying. As the ACP stands, video is processed a frame at a time or 30Hz. Flashing at this rate is not perceptible and the eye sees only an average value. The flashing problem is apparently in the structure of music, the normal audio input to the ACP. Music seems to have frequency changes the occur at a 3Hz rate, which is quite perceptible and in this case annoying. I won't speculate on why this is so for nothing simple can be said about it. What I can recommend though is that if the video processing rate is slowed down to say 1/2Hz that any flash produced will no longer be annoying for it will strobe at a slower rate.

Overall if it wasn't for the unexpected flashing effect the ACP would have been a total success. All the device functions met specifications except for a little non-linear clipping from the zener voltage window. Also the ACP introduced a small phase delay between the modified video color vectors. This caused some fringing but the effect was pleasing if anything at all.

# Conclusion

Using audio to modify video is still an intriguing idea. Unfortunately the actual implementation of this ACP was not the ideal approach. The biggest problem with the ACP is the flashing effect it produces. If this could be eliminated the ACP would be an aesthetic and interesting device.

One way of getting rid of the flashing is to replace the voltage adders with voltage multipliers. This way video that is high in intensity will have its intensity increased more than the low intensity colors. This would reduce the background flashing that is so annoying.

Although there is little use for the ACP as it stands a modification would produce a very useful device. The modification calls for a replacement of the adder/subtractor with a keyed switcher. When a key frequency is detected in a frame a switch is thrown to a new position for the duration of the frame. The switch can be used to bypass the vector or ground it or even switch in new video.

There are many applications for this effect. It can be used in a variety of processing applications and special effects generation. It could be used in the editing room to switch from one source to another or it could be used in a live camera shoot. Of course the ACP being an artist's tool, its only limitation of use is the creativity of its user.