Team 11

Varun Chadha, 2869615
Gage Phillips, 2878864
Noah Brabec, 2871029
Xinyun (Melody) Yu, 2870165

Project Name:

Retro Upscaler Box (R.U.B.)™

Project Synopsis:

Composite video upscaler for retro video game consoles, that improves image quality relative to linear interpolation used by modern high resolution displays.

Project Description:

The upscaler would allow users to plug in their game console, specifically targeting older consoles, and then from the upscaler to their tv. The upscaler will allow input of composite video signals and perform a conversion to digital formats for use in modern displays. There's a large number of retro game enthusiasts that wish to use their old consoles with modern, high resolution, large-format displays. Most scalers included with TVs will perform a linear interpolation of the video when the input is smaller than that TV's native resolution. This results in a blurry image that can be difficult to use, on top of just being subjectively dissatisfactory. This product will perform a scaling of the signal to the native resolution of the target display in a method that preserves sharpness. We are aiming to find a middle ground between raw signal processing and frame analysis to provide a more accurate fit for larger resolutions rather than just multiplying pixels (or performing a linear interpolation as current displays already do), and to provide minimal lag between console output and what is shown on the screen. Currently solutions in this field take either a direct frame analysis approach, resulting in large delay, or use direct signal manipulation which results in a product that is very fast, but can make sacrifices in exact target resolution and processing (such as deinterlacing).

Project Milestones:

First Semester

- September 27th:
  - Identify existing products
- October 4th:
  - Identify the weaknesses of the existing product
- October 11th:
  - Define the improvements we wish to seek
  - Begin search for hardware
- November 1st:
  - Device Block diagram
- November 8th:
  - Test existing algorithms
- November 22th

    - Acquire game consoles
- January 1st:
    - Explore run-time systems to use

Second Semester

- January 25th:
    - Explore FPGA hardware for the interpolation algorithm
- February 14th:
    - Assemble hardware
- March 14th:
    - Test hardware with video upscaling
- April 10th:
    - Refine upscaling algorithms
- May 15th:
    - Product casing design complete

Project Budget:

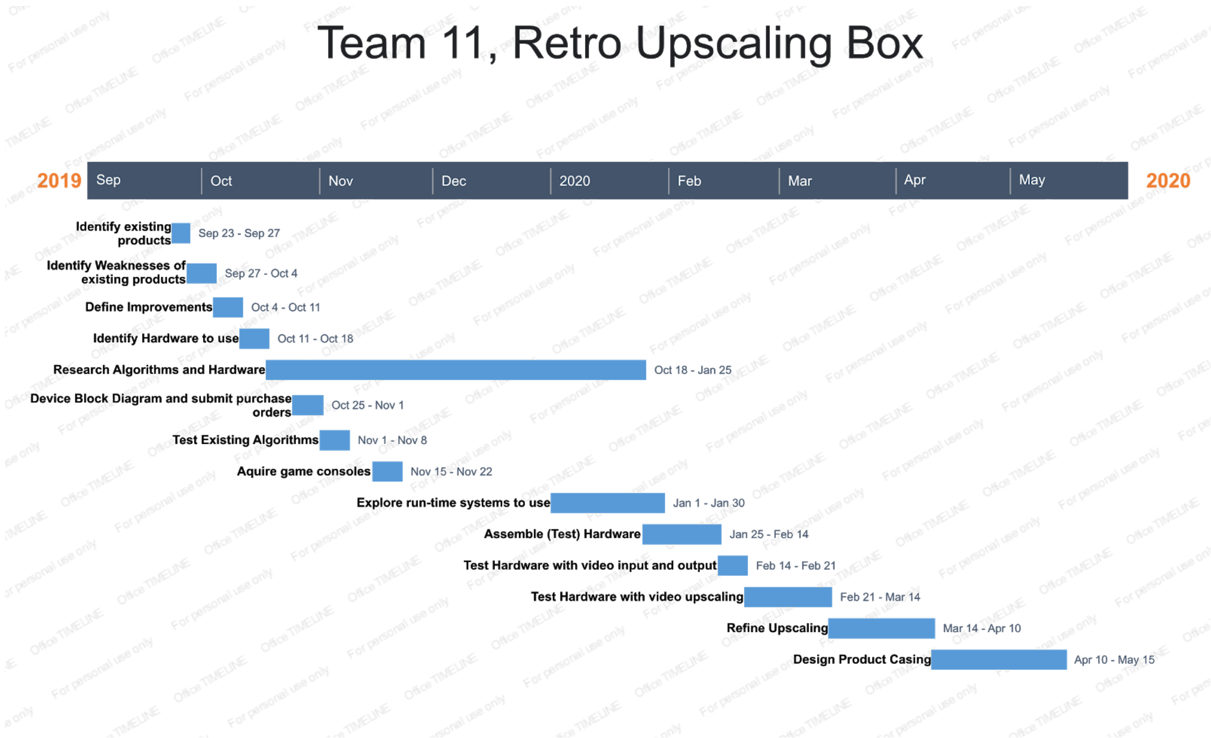
- (Free) Basys-3 FPGA Board, checked out from EECS Shop
- ($450) ZedBoard FPGA Development Board (or other HDMI-capable FPGA)
- ($150) Analog Devices ADV7280A Decoder Evaluation Board
- ($8) Cables
- (Provided by Members) Game Consoles.

*Some of these products can be found on popular marketplaces such as Amazon, but others like the FPGA boards could potentially be found from one of the preferred vendors such as CDW or Stanion Wholesale Electric. If the specific FPGA we wish for isn't available there, there is a chance we would need to purchase boards from a non preferred vendor such as Mouser Electronics or Digilent.
*We should not need too much training to work with these products due to everyone's experience with electronics from EECS 140 and EECS 388.

Work Plan:

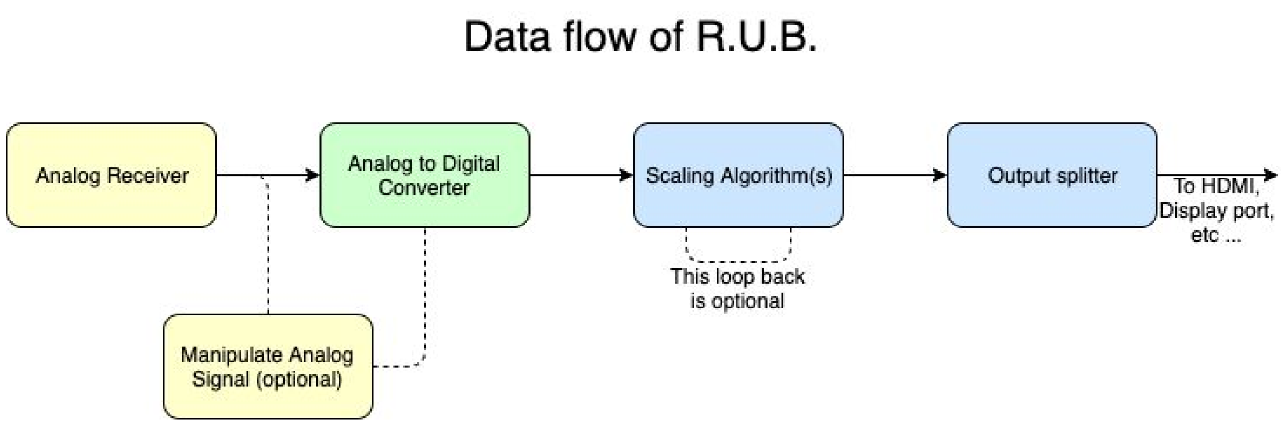
Many of the tasks we need are research or exploration based, so it'd be easy to have people do it on their own time, and then come together to share what they've learned. We could divide topics out amongst the members and try to have different members become some kind of relative subject-matter expert, but what we're looking for is a novel combination of ideas. To that end, the research and brainstorming processes are the responsibility of all members. However, there are some already solved problems that could be assigned, such as converting video and audio from the source composite format to some intermediary digital one, and then again into an HDMI output. As of now, it is likely that everyone will do a mild amount of research on the idea of video upscaling so we all have a good basis for the product idea. For the first semester, Varun will look into possible microcontrollers to utilize for the product. Gage will look into frame analysis and the conversion from an analog format to a digital one. Noah and Melody will also investigate frame analysis and possible output formats (HDMI). Varun will investigate hardware, and also investigate the conversion from composite signals. Additionally, Varun, Melody, and Gage will look into upscaling algorithms that could conform to our hardware restrictions. We were able to test the raspberry pi last semester and discovered that the read-in speed was too slow for capturing real time video data from the video decoder. So we decided to switch to the basys 3 board that could be checked out from the EECS shop. Noah and Gage will be testing and assembling the boards. Varun and Melody will look into the hardware based scaling algorithms.

Gantt Charts



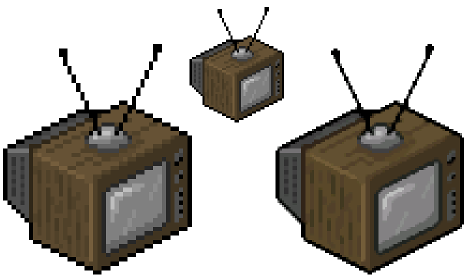
Preliminary Project Design:

How the software works:
Our solution to this scaling problem will result in an embedded system. We will lean on both hardware and software products to accomplish this. The hardware will consist of an FPGA that will be fast enough to perform the video processing since microcontrollers are not able to provide the required real-time performance. This FPGA will accept a digital signal and output HDMI. If we need to do any direct manipulation of the analog signal, it will be done at this stage before any kind of digital conversion. The analog video signal will be fed to a dedicated video decoder integrated circuit to allow for fast, efficient, and low cost conversion of the video signal from analog to digital.



This intermediary digital signal would then be used as input to the FPGA, where any processing of the digital video would happen. This is where the upscaling of the video would happen, and is also where any other software component would live. This could be any other optimizations like deinterlacing, or even some kind of basic runtime system with user-configurable options.

Traditionally, television displays utilize linear interpolation to scale content to the correct display size. However, this results in blurred images due to the algorithms effect of gradual transitions between pixel regions which is unwanted when upscaling pixel art. The algorithm we will choose to implement will create an image of the desired size, but without assuming anything about the “new” data. Side by side, the input and output images should look the same, as if the target display has the same native resolution as the game console is producing.



Two valid outputs derived from optional scaling algorithms.
https://en.wikipedia.org/wiki/Image_scaling#/media/File:2xsai_example.png

There is also a possibility to allow the use of multiple algorithms to achieve a greater quality of picture. The goal of this would be to allow a portion of the configuration to be pushed to the user, letting them add in or take out certain algorithms depending on their resolution needs.

Design constraints:

Technical Restraints:
Any electronic display capable of input from an HDMI signal will be capable of receiving the output signal from our product. We will aim to create the Retro Upscaler Box such that does as little augmentation to the original signal as possible, apart, of course, from the actual upscaling process. As long as our signal complies with HDMI specifications the output signal will not negatively affect the receiving electronic display.

Example of what tv scaling does vs our desired result.
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTobSy9Tyjsi_phsOGDcuhIKE06EZYmLClzvCM_EG3DQcSbuuoW&s

A gaming device that outputs a CVBS, or standard definition video, will be able to use our upscaler product. Provided that the input complies with either NTSC or PAL standards. The NTSC, Nation Television System Committee, standard is prevalent in North and Central America as well as Japan and South Korea. These regions are home to many of the companies that produce the most popular gaming devices. Therefore most video game signals from the most popular systems should be entirely compatible with our product. The PAL, Phase Alternating Line, is the system prefered in most of Europe, East and South Asia. Most of the hardware that we will ultimately employ can handle both NTSC and PAL standards.

In the design of our product, we will be bound by the video standards that are used to communicate video information. These signals are the language that televisions understand, so it is the language that we must speak. The older video standard that we (and many old game consoles) target will be composite video through an RCA physical interface. This is the input standard, but the output standard that we want to target is HDMI 1.3, a common digital display interface standard that has been around since 2006. If we found we had the throughput to output a 4K signal, we could instead use HDMI 1.4 which supports 4K video at 30Hz, and is otherwise backwards compatible with HDMI 1.3 displays.

Business Restraints:
Schedule - With 4 people on the team, each with a differing, compact schedule, it will be challenging to come up with a meeting time that works for everyone each week. In addition, due to holidays during the school year, where each of us will have personal plans, continuous meetings may be hard. Other than messaging through groupme, we could take advantage of video and meeting applications to have meetings online rather than in person.

Budget - Our specific budget constraints are dependent upon approval by Dr. Johnson. In the abstract, it will be important to keep total costs down based on the cost of individual components. Existing products that process composite signals and convert them into digital signals exist in a wide range of prices from as low as around $20 (for a simple converter) and up to $500(for a digital upscaler like the Retro Upscaler Box) or more. Clearly there exists an enormous disparity in the capabilities of these products. Our goal is to achieve some of the functionality of the higher end products while maintaining a reasonable cost. The prices of available products are lowered thanks to the economies of scale inherent in the manufacturing process. The development process of the Retro Upscaler Box will surely result in a cost higher than the cost of an actually manufactured product. To keep development costs as low as we can we will use available resources, provided by EECS, to assess the requirements that will ultimately be asked of the hardware. This will save us from having to expend our limited budget on a trial and error process to determine the hardware that will ultimately meet our expectations.

Team Composition and Makeup - Considering that our project involves embedded development and that the team consists of primarily software engineers, a consequential amount of research into video processing and FPGAs.

Software Licensing Restrictions/Requirements - At this point we cannot ascertain exactly what restrictions we may face regarding licensing. Though some hardware use is restricted to specific programming languages. When choosing the products we use during development this will need to be carefully considered. If some product has the specifications we believe we need it may require the use of a programming language we collectively have little familiarity with. This could slow down our development time.

Ethical Issues

- One issue that exists is that we are making a product that some people may be interested in buying for their own use. We don't plan on advertising it as such, but if we were to hand this off to others to use in exchange for a price, then we would want to make sure that the product delivers on all the promises that it makes. If we didn't do this, then we would effectively by lying to the consumers of our product and be in violation of section 1.3 of ACM's Code of Ethics by not being "honest and trustworthy".

Intellectual Property Issues

- One potential issue could be the use of licensed algorithms for upscaling the input video. If we use software that has a specific license that states it must not be modified or must be given reference to in the final product, we must respect these statements. However, if the license states that it cannot be used for commercial purposes then we must move on to another solution. If we do not respect the license of the software we utilize, we would lose our right to the software and we would be liable to civil tourt. To get around these issues, we could potentially write our own software.
- With our newfound use of FPGAs in our design, we can now utilize some pre-made VHDL code supplied by the manufacturer to perform certain tasks, like creating a VGA signal or converting color spaces. These solutions also have their own licenses that must be abided by. We would be starting with an evaluation license, which we would be able to use to test and develop the product for a limited amount of time. If this were to be developed into an actual product we would need to get an actual license, but for our testing purposes the evaluation license should do fine.

Change Log

- Milestones updated - We modified and also added some new milestones to our planner as we look into more resources and have a better understanding of what we should do. We increased the resolution of tasks so as to balance the workload for this and next semester as well as to make the goals more well defined. These changes are as followed:
    - We have greatly increased the time for researching and looking for hardware
    - We have pushed the assembly and testing of hardware back to the middle of the second semester
    - Additionally the refining of scaling algorithms has also been push back to match the assembly of hardware timeline
- Work Plan refined - The work plan assigned to everybody in the initial project description stays the same. In addition, we will need to research and decide on an algorithm to use for upscaling the video, which has now been added to the current work plan for Varun, Gage and Melody.
- We updated the document to reflect the loss of our team member, Joe Goldsiech
- Updated document to replace the use of microcontrollers with FPGAs
- Added pre-made FPGA components to the Intellectual Property Issues section
- Updated budget to include FPGA Boards