Capstone Project Proposal

RISC-V Secure Hardware Video Output

Authors:

Ahmad Alothaimin, Hector Soto, Jack Chen,

Ross Wegter, Ryan Nand

January 21, 2021

Revision 0.1.0

Table of Contents

Project Design Specification **3**

- Background 3
- Overview / Executive Summary 3
- Concept of Operations 4
- Stakeholders 4
- Requirements 5
- Specifications 5
- Deliverables 5
- Initial Product Design 6
- Verification Plans 8

Project Management Plan **9**

- Timeline, with milestones 9
- Budget and Resources 12
- Team and Development Process 13
  - Team and Skills 13
  - Roles and Responsibilities 13
  - Collaboration Tools 14
  - Methodology 14

References **15**

Project Design Specification

Overview

Our industry sponsor is Galois, and we are implementing a video output onto their RISC-V secure hardware. This implementation would be a part of their Balancing Evaluation of System Security Properties with Industrial Needs (BESSPIN) project which is funded by System Security Integration Through Hardware and Firmware (SSITH), a program under Defense Advanced Research Projects Agency (DARPA). The project involves a Galois' SoC implementation on a VCU118 FPGA development board that incorporates novel computer security features.

Currently the only way to interact with and obtain information on the System on Chip (SoC) is over a serial connection. To better demonstrate the security capabilities of the RISC-V based SoC, as well as to help Galois developers on the BESSPIN project expedite the information gathering and debugging process, having a video signal output on the SoC, which may be used to generate a graphical user interface, is desired.

Our objective is to allow the SoC to have video output capabilities on the VCU118. This will involve changes to the firmware on the FPGA as well as software changes to the RISC-V based Linux kernel located on the SoC. By the end of this project, we must be able to plug in a standard video signal cable (such as an HDMI or VGA cable) onto the VCU118 connected to a monitor and the monitor must display something. The display must be able to show an image or video and should be a terminal window. We plan to develop this product through Agile-based methods through incremental progress and weekly check-ins with Galois.

Background

Flaws in software design for electronic systems can enable hackers to obtain privileged access rights, allowing undesired alteration of those systems or acquisitions of private data. Software updates can't always address this issue, which is why looking at hardware instead as an approach could provide electronic systems better security against cyberthreats.

The Department of Defense (DoD) is concerned with mitigating such attacks and started a program under DARPA to research and develop hardware security architectures and design tools under the program SSITH[2].

Thus far, DARPA is successful in proving that this approach works by crowdsourcing hackers to attempt to break into 5 secure processors developed by organizations on the SSITH program through the Finding Exploits to Thwart Tampering (FETT) 3 month long bug bounty program[4]. While no end result has been publicly published yet, the state of this program had no secure processors be compromised for the first half of the program[3].

Participating in the SSITH program is Galois, a technology research company. Galois has developed novel evaluation tools for these hardware architectures with security purposes through their Balancing Evaluation of System Security Properties with Industrial Needs (BESSPIN) project. Evaluation is done with an SoC to examine other architectures developed for the SSITH program.

Approach

The SoC can be implemented to have video output capability as part of its firmware on the FPGA. This video capability would accept information (some examples may be a frame, some text, or an image) to be displayed from the Linux kernel being run on the SoC. This video capability would then output a valid video signal onto a standard video signal cable (examples include VGA or HDMI) to display the received information onto a monitor.

The main stakeholders for this capstone project are Galois' developers and DARPA employees working on the BESSPIN and SSITH projects. These stakeholders want to more easily demonstrate the novel security capabilities that the SoC on the VCU118 has for the BESSPIN project through a visual medium via video capability on the device. This video capability will be done by modifying the hardware of the device, with possible software changes to allow the current SoC to interact with the video capabilities. After development, this capability to show video will be given to and used by Galois' developers through detailed setup instructions, access to needed hardware (VCU118 and a connection to a monitor via a standard video signal cable such as HDMI), and access to a repository that includes needed software.

Because this project is programming-based, this solution can be reused and can be scalable for the nuances of video signals. If successful, the project will physically look like the development board (VCU118) connected to a monitor through some form of video signal cable and digitally appear as a combination of Hardware Description Languages (HDLs) and potentially Linux kernel modules for any possible software changes. Both the dev board and monitor will be powered via wall outlets and when all components are connected and powered, graphics will appear on the monitor.

Stakeholders

The following are the stakeholders in this project:

- Galois developers on the BESSPIN project
- DARPA employees under the SSITH program
- Portland State University Students in ECE Capstone Team 7

Requirements

Musts

- We must have signed an NDA which covers non-public information about the SoC.
- We must change the firmware on the SoC on the VCU118 FPGA development board to output an image to a monitor.
- We must demonstrate the output by Friday, May 28th.
- We must develop and push any changes to the project onto the `feature/video-output` branch in the gfe repository.
- We must document our development process.

Shoulds

- We should be able to display a terminal from the Linux kernel over our video output capability.
- We should have weekly meetings to go over the direction and address issues during the project.

Mays

- We may be able to display a desktop environment.
- We may implement multiple video output methods (VGA, DVI-A, DVI-D, PCIe to GPU, etc.)
- We may have dynamic resolution for the video output.

Specifications

- Modified SoC hardware capable of video output within the system's 100 MHz clock speed.
- Modified software (Linux or FreeBSD drivers) is able to provide video output through the hardware.
- Demonstration program showing video output capability of the SoC.

Deliverables

In addition to this project proposal our deliverables to Galois will include weekly progress reports via video conferencing, detailed design documentation, version control, Intellectual Property information, a working demonstration of the video output, a final detailed project report, and an ECE Capstone Poster Session poster. Detailed design documentation will explain what our design does, how our design works, and why we made specific design decisions. Version control will include checked-in previous revisions of our design via changes to the GFE repo provided by Galois. The final project report will have specific and detailed technical (as well as

quantitative) information that is relevant to the project with an accompanying poster to help further understanding as a visual aid summary. All project documentation deliverables will be finalized and submitted to Galois no later than June 11th 2021.

Initial Product Design

The HDL code and possibly linux kernel changes we will provide our sponsor implement a means to display picture elements from DRAM on the VCU118 to a monitor connected to the PMOD connectors of the same VCU118. We will modify Galois' GFE VCU118 remotely through Vivado design suite by adding our own Display IP to the FPGA which may be a combination of Xilinx IP modules. Also, we will attach a PMOD DVI board to the PMOD connectors of the VCU118 as a T.M.D.S. transmitter to output picture elements over HDMI to a connected monitor.

A large risk of this project is remotely working on the VCU118. This includes development on the FPGA, physical measurement of the PMOD connectors, and testing methods. Development on the FPGA will be a risk as building a bit stream will require host systems with enough system memory (suggested 32GB) to complete in a reasonable amount of time. The dual PMOD connectors on the VCU118 do not align exactly with the PMOD DVI board and a solution to this will be difficult to find while working remotely. Lastly, not being able to see visual representation of our output to the monitor after each iteration could be a large cause for concern. Possibly the quickest resolutions to these issues will be direct communication with those physically at Galois near the VCU118. However, we will also develop a means to test expected output to actual output through a simulation testbench.

We will still need to determine if we will be able to use the Linux kernel framebuffer as a means to stream picture elements. This determination will come through further research and testing. However, we feel confident in completing what we propose in this document within a 5-month period. Below are level 0 diagrams of our proposed solution.

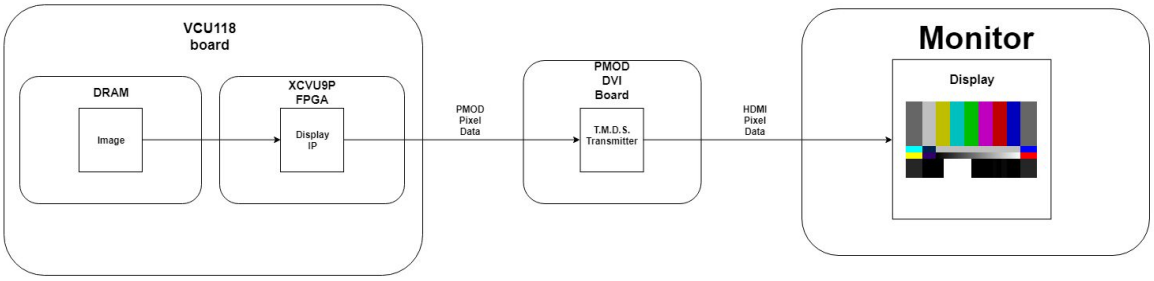


Figure 1: Overview Level 0 Diagram

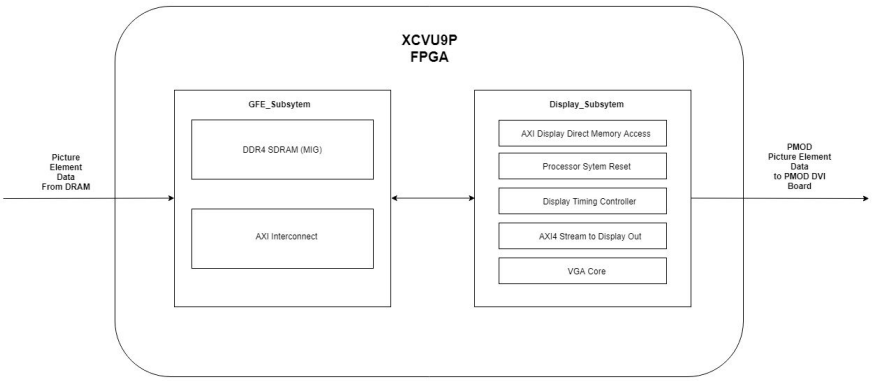


Figure 2: FPGA Level 0 Diagram

For the software development, we will primarily use the Vivado Design Suite Environment. The primary languages we will use are Verilog and C though some VHDL or Python could be used. The end-user will use our product by simply connecting a powered monitor via hdmi to the PMOD DVI board after having setup and powered the VCU118. Once all connections are made, an image will be displayed on the monitor. Also, the user can access the image code on the VCU118 through Vivado design suite.

If we hit a major roadblock when trying to incorporate the 12bit PMOD DVI board, we will adjust and use the 3bit PMOD DVI board. Also, if we find that the PMOD DVI boards are not a viable option we will switch to using PMOD VGA boards as a backup as this is the closest option in terms of picture element to display processing methods. Only when all of those options fail will we turn to using a graphics card to transmit picture elements from the VCUs PCIe connection.

Verification Plans

We know the product will be working if the intended image is displayed on a monitor with the source of that image being the DRAM of the connected VCU118. To verify all needed connections are made and picture elements are transmitted as intended we will test both hardware and software.

For hardware, we will test each component of our solution. This starts by testing the FPGA's PMOD connector and making sure that it is correctly connected and compatible with the FPGA. Then, we proceed by testing the DVI cable and the monitor to see if our output is functional.

For software testing, we will make sure that our codes meet the requirements of our solution. Our code will be using Video Frame Buffer Read and Video Frame Buffer Write IP from Xilinx. Testing the code will involve testing our generic inputs for DVI (clk, red, blue, green,

active, hsync, vsync), user ports, memory read data, the timing core which will output the generic outputs (colors) when our active signal is high, state machines, and algorithm. We will develop a testing plan as we write our codes to ensure that our product meets specification.

Project Management Plan

Timeline, with milestones

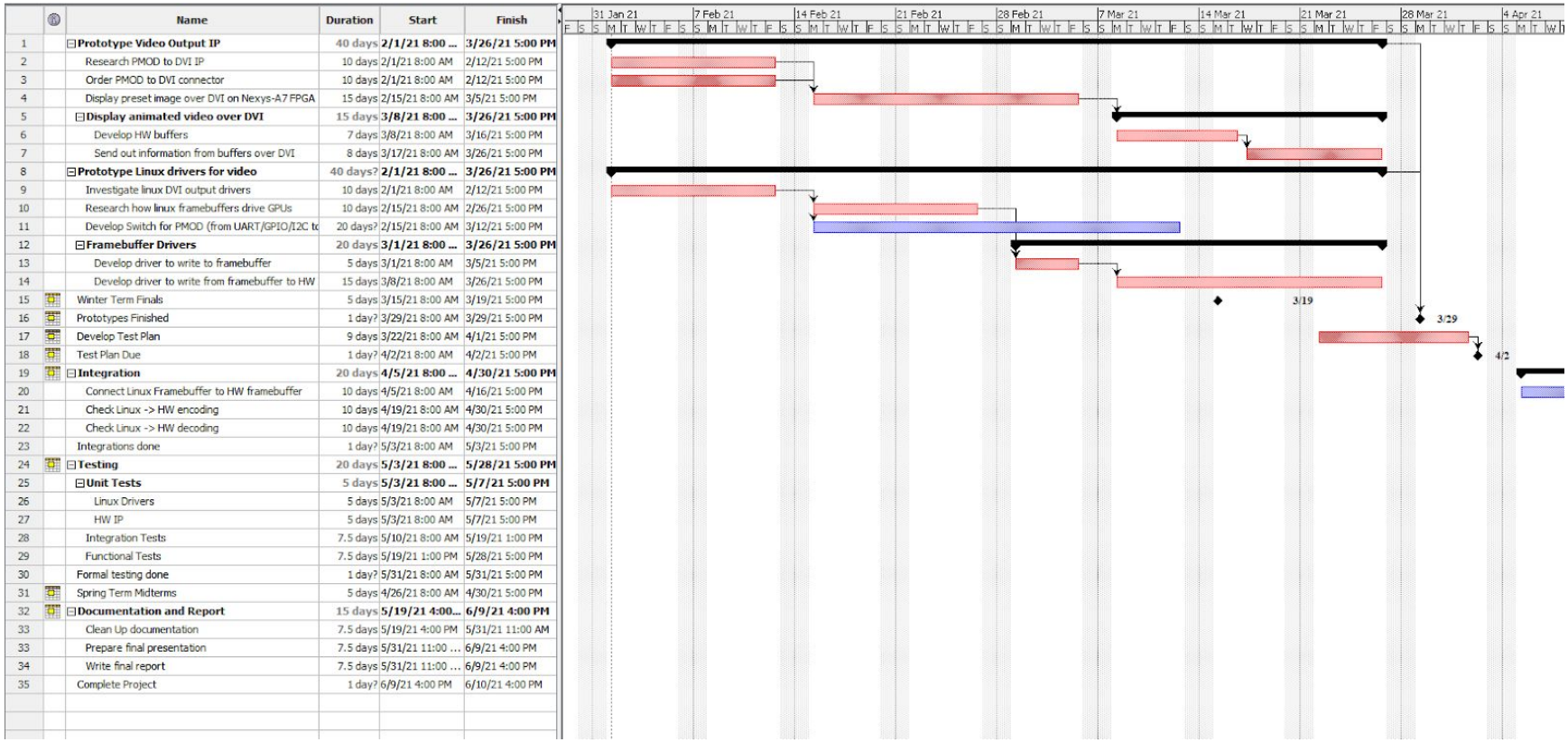
Our general timeline is the following:

Timeline	Process
February - mid March	Start prototyping
Mid March - April	Make test plan
April - May	Integration
Mid May - June	Documentation and final report

Table 1: Table of Timeline

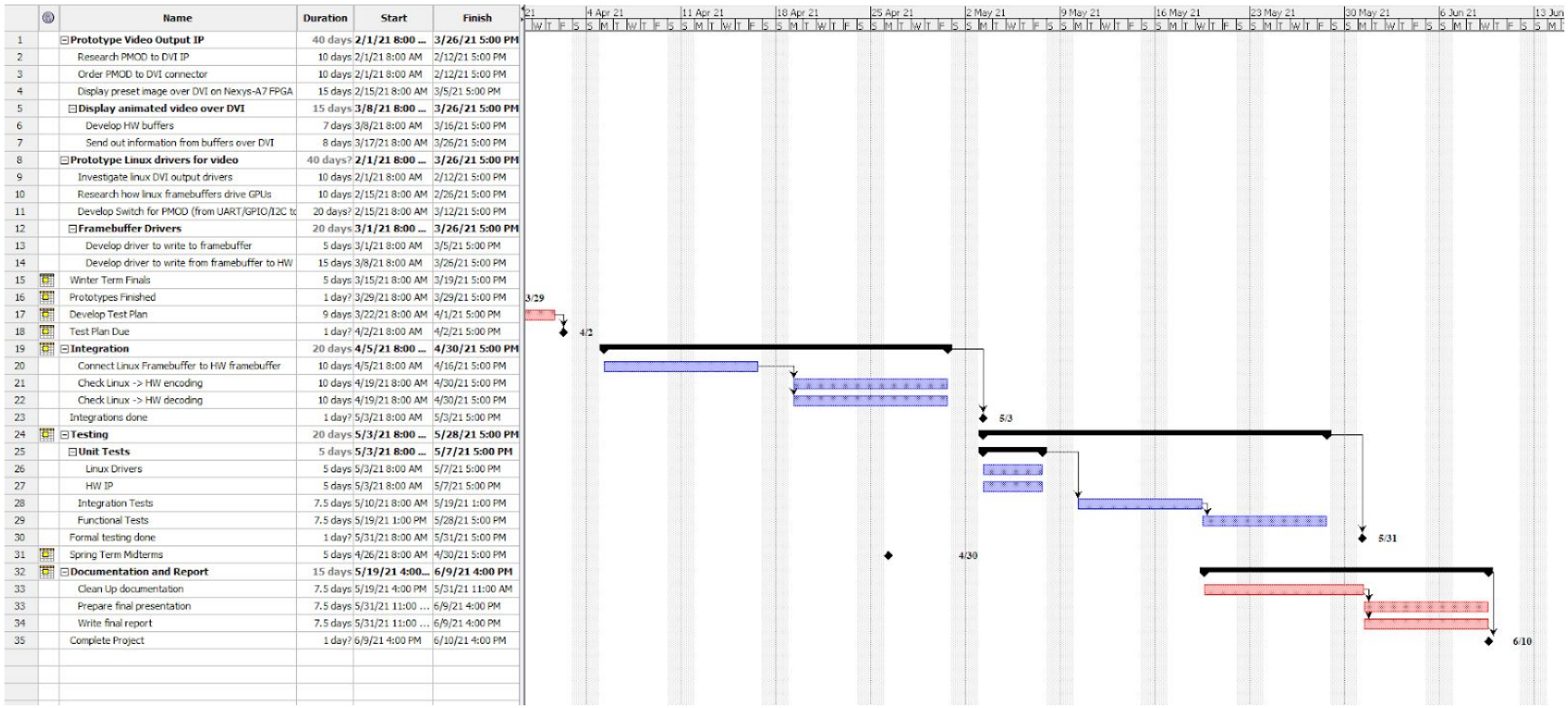
This schedule predicts an initial phase of prototyping in parallel; one path devoted for development of the hardware IP, another path for Linux drivers and kernel development. These paths are tentative and more time than expected is added to the duration of each task. Simple milestones are added at the end of each major of the tasks and are listed on the Gantt chart as dates.

At the beginning of each week, we will check-ins with Galois with exceptions due to vacations, exams, or other circumstances communicated over email. This check-in would address any questions or roadblocks during development as well as showing off any weekly incremental progress.



	Name	Duration	Start	Finish
1	**Prototype Video Output IP**	40 days	**2/1/21 8:00 ...**	**3/26/21 5:00 PM**
2	Research PMOD to DVI IP	10 days	2/1/21 8:00 AM	2/12/21 5:00 PM
3	Order PMOD to DVI connector	10 days	2/1/21 8:00 AM	2/12/21 5:00 PM
4	Display preset image over DVI on Nexys-A7 FPGA	15 days	2/15/21 8:00 AM	3/5/21 5:00 PM
5	**Display animated video over DVI**	15 days	**3/8/21 8:00 ...**	**3/26/21 5:00 PM**
6	Develop HW buffers	7 days	3/8/21 8:00 AM	3/16/21 5:00 PM
7	Send out information from buffers over DVI	8 days	3/17/21 8:00 AM	3/26/21 5:00 PM
8	**Prototype Linux drivers for video**	40 days?	**2/1/21 8:00 ...**	**3/26/21 5:00 PM**
9	Investigate linux DVI output drivers	10 days	2/1/21 8:00 AM	2/12/21 5:00 PM
10	Research how linux framebuffers drive GPUs	10 days	2/15/21 8:00 AM	2/26/21 5:00 PM
11	Develop Switch for PMOD (from UART/GPIO/I2C to	20 days?	2/15/21 8:00 AM	3/12/21 5:00 PM
12	**Framebuffer Drivers**	20 days	**3/1/21 8:00 ...**	**3/26/21 5:00 PM**
13	Develop driver to write to framebuffer	5 days	3/1/21 8:00 AM	3/5/21 5:00 PM
14	Develop driver to write from framebuffer to HW	15 days	3/8/21 8:00 AM	3/26/21 5:00 PM
15	Winter Term Finals	5 days	3/15/21 8:00 AM	3/19/21 5:00 PM
16	Prototypes Finished	1 day?	3/29/21 8:00 AM	3/29/21 5:00 PM
17	Develop Test Plan	9 days	3/22/21 8:00 AM	4/1/21 5:00 PM
18	Test Plan Due	1 day?	4/2/21 8:00 AM	4/2/21 5:00 PM
19	**Integration**	20 days	**4/5/21 8:00 ...**	**4/30/21 5:00 PM**
20	Connect Linux Framebuffer to HW framebuffer	10 days	4/5/21 8:00 AM	4/16/21 5:00 PM
21	Check Linux -> HW encoding	10 days	4/19/21 8:00 AM	4/30/21 5:00 PM
22	Check Linux -> HW decoding	10 days	4/19/21 8:00 AM	4/30/21 5:00 PM
23	Integrations done	1 day?	5/3/21 8:00 AM	5/3/21 5:00 PM
24	**Testing**	20 days	**5/3/21 8:00 ...**	**5/28/21 5:00 PM**
25	**Unit Tests**	5 days	**5/3/21 8:00 ...**	**5/7/21 5:00 PM**
26	Linux Drivers	5 days	5/3/21 8:00 AM	5/7/21 5:00 PM
27	HW IP	5 days	5/3/21 8:00 AM	5/7/21 5:00 PM
28	Integration Tests	7.5 days	5/10/21 8:00 AM	5/19/21 1:00 PM
29	Functional Tests	7.5 days	5/19/21 1:00 PM	5/28/21 5:00 PM
30	Formal testing done	1 day?	5/31/21 8:00 AM	5/31/21 5:00 PM
31	Spring Term Midterms	5 days	4/26/21 8:00 AM	4/30/21 5:00 PM
32	**Documentation and Report**	15 days	**5/19/21 4:00...**	**6/9/21 4:00 PM**
33	Clean Up documentation	7.5 days	5/19/21 4:00 PM	5/31/21 11:00 AM
33	Prepare final presentation	7.5 days	5/31/21 11:00 ...	6/9/21 4:00 PM
34	Write final report	7.5 days	5/31/21 11:00 ...	6/9/21 4:00 PM
35	Complete Project	1 day?	6/9/21 4:00 PM	6/10/21 4:00 PM

Image 1: Project Timeline before April 2nd Milestone of Completed Test Plan



	Name	Duration	Start	Finish
1	**Prototype Video Output IP**	40 days	**2/1/21 8:00 ...**	**3/26/21 5:00 PM**
2	Research PMOD to DVI IP	10 days	2/1/21 8:00 AM	2/12/21 5:00 PM
3	Order PMOD to DVI connector	10 days	2/1/21 8:00 AM	2/12/21 5:00 PM
4	Display preset image over DVI on Nexys-A7 FPGA	15 days	2/15/21 8:00 AM	3/5/21 5:00 PM
5	**Display animated video over DVI**	15 days	**3/8/21 8:00 ...**	**3/26/21 5:00 PM**
6	Develop HW buffers	7 days	3/8/21 8:00 AM	3/16/21 5:00 PM
7	Send out information from buffers over DVI	8 days	3/17/21 8:00 AM	3/26/21 5:00 PM
8	**Prototype Linux drivers for video**	40 days?	**2/1/21 8:00 ...**	**3/26/21 5:00 PM**
9	Investigate linux DVI output drivers	10 days	2/1/21 8:00 AM	2/12/21 5:00 PM
10	Research how linux framebuffers drive GPUs	10 days	2/15/21 8:00 AM	2/26/21 5:00 PM
11	Develop Switch for PMOD (from UART/GPIO/I2C to	20 days?	2/15/21 8:00 AM	3/12/21 5:00 PM
12	**Framebuffer Drivers**	20 days	**3/1/21 8:00 ...**	**3/26/21 5:00 PM**
13	Develop driver to write to framebuffer	5 days	3/1/21 8:00 AM	3/5/21 5:00 PM
14	Develop driver to write from framebuffer to HW	15 days	3/8/21 8:00 AM	3/26/21 5:00 PM
15	Winter Term Finals	5 days	3/15/21 8:00 AM	3/19/21 5:00 PM
16	Prototypes Finished	1 day?	3/29/21 8:00 AM	3/29/21 5:00 PM
17	Develop Test Plan	9 days	3/22/21 8:00 AM	4/1/21 5:00 PM
18	Test Plan Due	1 day?	4/2/21 8:00 AM	4/2/21 5:00 PM
19	**Integration**	20 days	**4/5/21 8:00 ...**	**4/30/21 5:00 PM**
20	Connect Linux Framebuffer to HW framebuffer	10 days	4/5/21 8:00 AM	4/16/21 5:00 PM
21	Check Linux -> HW encoding	10 days	4/19/21 8:00 AM	4/30/21 5:00 PM
22	Check Linux -> HW decoding	10 days	4/19/21 8:00 AM	4/30/21 5:00 PM
23	Integrations done	1 day?	5/3/21 8:00 AM	5/3/21 5:00 PM
24	**Testing**	20 days	**5/3/21 8:00 ...**	**5/28/21 5:00 PM**
25	**Unit Tests**	5 days	**5/3/21 8:00 ...**	**5/7/21 5:00 PM**
26	Linux Drivers	5 days	5/3/21 8:00 AM	5/7/21 5:00 PM
27	HW IP	5 days	5/3/21 8:00 AM	5/7/21 5:00 PM
28	Integration Tests	7.5 days	5/10/21 8:00 AM	5/19/21 1:00 PM
29	Functional Tests	7.5 days	5/19/21 1:00 PM	5/28/21 5:00 PM
30	Formal testing done	1 day?	5/31/21 8:00 AM	5/31/21 5:00 PM
31	Spring Term Midterms	5 days	4/26/21 8:00 AM	4/30/21 5:00 PM
32	**Documentation and Report**	15 days	**5/19/21 4:00...**	**6/9/21 4:00 PM**
33	Clean Up documentation	7.5 days	5/19/21 4:00 PM	5/31/21 11:00 AM
33	Prepare final presentation	7.5 days	5/31/21 11:00 ...	6/9/21 4:00 PM
34	Write final report	7.5 days	5/31/21 11:00 ...	6/9/21 4:00 PM
35	Complete Project	1 day?	6/9/21 4:00 PM	6/10/21 4:00 PM

Image 2: Project Timeline from April 2nd to Jun 10th Milestone of Complete Project

Budget and Resources

Not all of the list resources may be needed or used depending on how our development progresses and will be obtained by request after acknowledgement from all parties.

Item	Description	Cost	Source
Access to repository	The repository contains source files and build scripts for the SoC	Provided by sponsors	Galois
Vivado license	Required to use Xilinx Vivado software suite	Provided by sponsors	Galois
Remote access to FPGA	Required to access and modify Vivado project	Provided by sponsors	Galois
PMOD to DVI or PMOD to VGA	Used to add video output to FPGA	$16.95	https://1bitsquared.com/products/pmod-digital-video-interface
HDMI cable	Cable used to connect the PMOD DVI to a monitor	Provided by sponsors	Galois
Monitor	Required to show output from SoC	Provided by sponsors	Galois
Development environments	Allowing to rapidly prototype designs remotely	Provided by us	Team members
PCIe x16 Female to Female Riser	Allows the connection of the VCU118 PCIe male riser to a GPU	$27.99	https://www.amazon.com/Express-Flexible-Extension-Gold-Plated-Connector/dp/B07PMRBXXG
GPU	A PCIe to GPU connection may be needed. If so, a GPU is required	~$10-$100	Many https://www.ebay.com/str/freegeekbasicsstore

Table 2: Resources Table

Team and Development Process

Team and Skills

Ross Wegter

- Embedded Systems Engineering
- Linux Driver Development
- Software Engineering with a focus on Object Oriented Design and Data Structures
- Verbal and Written Communication
- Teamwork and Management

Ahmad Alothaimin

- Verilog/SystemVerilog
- C
- Verification of hardware and software
- Communication

Hector Soto

- Coding: C/C++/C#, Assembly (ARM + AVR), SystemVerilog, MATLAB
- Design: Circuit Design (LTspice), PCB Design (KiCad)
- 3D modeling (OnShape) and printing
- Soldering

Jack Chen

- Image Processing and Machine Learning Researcher
- Xilinx Vivado Development of FPGAs using SystemVerilog
- Student Leader and Manager of Annual Conference
- C/C++, Python, ARM Assembly

Ryan Nand

- Verilog
- C/C++

Roles and Responsibilities

Initially, each of us will be assigned to a subproject explained down below in the methodology section. Of course, we will be assigned to different tasks within each subproject and these tasks may change throughout development. Additionally, we may jump around to help those of us that may not be progressing as fast as the project requires.

We have dedicated Ahmad Alothaimin as the middle-man to communicate with the industry sponsor and the faculty advisor. This may help reduce and/or eliminate any confusion and unnecessary actions made by us.

We have elected Jack Chen for the role of team manager. This role is more so for keeping tasks in Trello organized and making sure the members are focused on their individual tasks.

Collaboration Tools

For internal team communication we have established a Discord channel. This is a much better alternative to frequent back and forth emails in a cluttered inbox. In addition, this enables us to have quick and easy access to video chat.

All faculty and sponsor meetings will be conducted over Zoom. This way there will be an option for recording the meetings if the need arises.

For internal task management, we have established a Trello board. Any of us can update what we are working on or what we should be working on by looking at this board.

We will be using github wiki for the location of all our technical documentation from the project.

Methodology

Our initial method is to break the project into hardware and software subprojects and then assign each of us to attack these subprojects. These subprojects will be handled using methodologies from the spiral project management. There will be numerous reviews and analyses of the solutions to these subprojects. These reviews will not only be conducted by us but by the industry sponsor and the faculty advisor.

References

[1] D. Zimmerman and M. Podhradsky, “RISC-V Secure Hardware Video Output,” Available: http://web.cecs.pdx.edu/~faustm/capstone/projectdescriptions/2021/cd5cb5a5-62ab-4e60-b976-045751974f5b/ECE%20Capstone%20RISCV%20secure%20hardware%20video%20output%20.pdf [Accessed Jan. 22, 2021]

[2] K. Rebello, “System Security Integration Through Hardware and Firmware (SSITH),” Available: https://www.darpa.mil/program/ssith [Accessed Jan. 22, 2021]

[3] L. C. Williams, “DARPA's new hardware proves tough to crack,” *Federal Computer Week,* 24 Aug. 2020. Available: https://fcw.com/articles/2020/08/24/williams-darpa-cyber-hardware.aspx

[4] DARPA, “DARPA FETT Bug Bounty Program,” Available: https://fett.darpa.mil/