

# **2-D Platform Control Using an FPGA**

Final Report  
May 05-22

Client:  
Dr. Mani Mina

Faculty Advisor:  
Dr. Mani Mina

Senior Design Team:  
Dillon Glissman, CPRE  
Cipto Kurniawan, EE  
Clinton Middaugh, CPRE  
Mark Truckenbrod, EE

DISCLAIMER: This document was developed as a part of the requirements of an electrical and computer engineering course at Iowa State University, Ames, Iowa. This document does not constitute a professional engineering design or a professional land surveying document. Although the information is intended to be accurate, the associated students, faculty, and Iowa State University make no claims, promises, or guarantees about the accuracy, completeness, quality, or adequacy of the information. The user of this document shall ensure that any such use does not violate any laws with regard to professional licensing and certification requirements. This use includes any work resulting from this student-prepared document that is required to be under the responsible charge of a licensed engineer or surveyor. This document is copyrighted by the students who produced this document and the associated faculty advisors. No part may be reproduced without the written permission of the senior design course coordinator.

1 April 2005

## **Table of Contents**

List of Tables.....	3
List of Figures.....	4
List of Definitions.....	4
<b>Part 1: Introductory Material.....</b>	<b>6</b>
1.1 Executive Summary.....	6
1.2 Acknowledgement.....	7
1.3 Problem Statement.....	8
1.4 Operating Environment.....	8
1.5 Intended User(s) and Use(s).....	8
1.6 Assumptions and Limitations .....	8
1.7 Expected End Product and Other Deliverables.....	9
<b>Part 2: Analysis of Hardware and Software Approach.....</b>	<b>10</b>
2.1 Software Approach.....	10
2.2.The Hardware Approach.....	16
<b>Part 3 Resources and Schedules.....</b>	<b>21</b>
3.1 Personnel effort.....	21
3.2 Financial Resources.....	22
3.3 Schedules.....	23
<b>Part 4: Closure Materials.....</b>	<b>25</b>
4.1 Product Evaluation.....	25
4.2Commercialization.....	25
4.3 Recommendations for Additional Work.....	25
4.4 Lessons Learned.....	26
4.5 Risk and Risk Management.....	26
<b>Part 5: Contact Information.....</b>	<b>28</b>
5.1 Client Information.....	28
5.2 Faculty Advisor Information.....	28
5.3 Project Team Information.....	28
5.4 Closing Summary.....	29
<b>Part 6: References.....</b>	<b>30</b>
<b>Part 7: Appendices.....</b>	<b>31</b>

## **Tables**

Table 1: Wiring to the SCB-68.....	19
Table 2: Revised Personnel Hours.....	21
Table 3: Actual Personnel Hours.....	22
Table 4: Revised Budget.....	23
Table 5: Actual Budget.....	23

## **Figures**

Figure 1: FPGA Software Design.....	11
Figure 2: LabVIEW VI Software Design.....	12
Figure 3: FPGA code static values and front end control.....	14
Figure 4: Temporary front end for the FPGA.....	15
Figure 5: Stepper Motor Driver.....	16
Figure 6: Scanner.....	17
Figure 7: SCB-68 Boa.....	17
Figure 8: PK3 Drive Pins.....	18
Figure 9: QM Motor Wiring.....	18
Figure 10: FPGA Card.....	18
Figure 11: NI PXI.....	18
Figure 12: System Flow Diagram.....	19
Figure 13: Original Personnel Hours.....	21
Figure 14: Initial Budget.....	22
Figure 15: Initial Gant Chart.....	23
Figure 16: Revised Gant Chart.....	24
Figure 17: Actual Gant Chart.....	24
Figure 18: Deliverables Schedule.....	25

## **List of Definitions**

The following are definitions of terms used in the project:

- FPGA (field programmable gate array)- Programmable logic device
- Programmable logic device- controller that is programmable by the end user of the device
- Precision cutting device- device that cuts within a specified tolerance
- Instruction- One of nine possible directions in two-dimensional space that the laser can be moved in. Nine dimensions will be in two- dimensional space, including staying at the origin as a possible direction. A five bit signal will represent each instruction.
- Prototype- When the word prototype is used in the report, it is a reference to the system as a whole that is being designed to have precise user controlled scanning in two dimensions. This includes, but is not limited to a scanner, stepper motors, stepper motor drivers, FPGA, and PXI Computer. This also includes all LabVIEW VI software designed to control the above listed hardware.
- CSG – Computer Support Group of the Department of Electrical and Computer Engineering at Iowa State.

## **Part 1 Introductory Material**

### **1.1 Executive Summary**

Precise and steady control of a scanner can be difficult to obtain. The laser will clearly show if the scanner moves past the desired position, wobbles, or cannot trace the same path twice. The May 05-22 Senior Design team will attempt demonstrate precise control of scanner movement over a range user chosen shapes, using a field programmable gate array and a scanner system.

The objective of this report is to track the final progress of the May 05-22 design team. The report includes lists of operating environment needs, intended uses, intended users, assumptions, limitations, and expected deliverables of the project, as well as recommendations on continuing the project and guidance for the next Senior Design team if the project is to continue.

The report includes the design requirements, functional requirements, design constraints, technical and testing approaches.

The use and benefits of LabVIEW Virtual Instrument software, and the software design for the National Instruments PXI computer chassis, and a National Instruments 7831R FPGA using the LabVIEW Virtual Instrument language were explored. The various processing steps and the path of data flow controlling the selected scanner hardware are shown, both the interaction from PXI to the FPGA and from interaction from the FPGA to the hardware and the PXI. In addition, an alternate solution to the data flow between the PXI to the FPGA is shown.

All major hardware components for the project and their uses are discussed, including: a Parker Precision Scanner, Compumotor QM style digital stepper motors, Compumotor PK3 stepper motor controls, National Instruments SCB-68 Shielded Connector Block, National Instruments 7831R FPGA, and National Instruments PXI computer chassis. The wiring schematic from the motors to the stepper driver is shown, and the inputs to control the stepper motor are also shown.

Resources utilization, which includes personnel management and projected work hours for each team member, various resources used, and projected financial resources for the project are examined. The scheduling of project tasks and the project deliverable scheduling are also shown.

The report is concluded with contact information of the client, advisor, and team members, a brief summary of the project, a list of reference material used in the project, and attached appendices that apply to the project.

## **1.2 Acknowledgement**

The group would like to acknowledge Dr. Mani Mina for providing guidance to the group as the client and project advisor. The group would also like to acknowledge National Instruments for the donation of parts for the project.

### **1.3 Problem Statement**

Precise and steady control of a scanner can be difficult to obtain. A laser will clearly show if the scanner moves past the desired position, wobbles, or cannot trace the same path twice. Once steady scanner movement is obtained, difficulty can also arise in the mathematical and programming methods needed to allow users to select and modify various shapes for precise scanning. The goal of the group is to develop a hardware and software prototype system to solve these problems, based on existing hardware and the LabVIEW Virtual Instrument hardware language.

### **1.4 Operating Environment**

The prototype designed by the group will work in a climate-controlled environment. The PK3 stepper motor drivers have the following qualifications for operation and storage:

- Relative Humidity will range from 0% to 95% (non-condensing)
- Storage temperature will be within the range of -40 to 85 degrees Celsius
- Operational temperature will range from 0 to 40 degrees Celsius. (32 to 102 degrees Fahrenheit)

### **1.5 Intended User(s) and Use(s)**

The following are intended users and uses for the project.

#### **1.5.1 Intended Users**

Intended users include the programmer and operator. It is recommended that the programmer is at least 21 years old and in good physical and mental health. He or she will define the operating parameters and will need some technical background specifically in LabVIEW environment. The intended operator should be at least 18 years old, with ability to handle things with care. He or she will make sure the machine is functioning as expected and will need minimal training for the device. A minimum of a high school education is recommended for the operator.

#### **1.5.2 Intended Uses**

Intended uses for the project is for precision scanning in manufacturing area with precision. The implementations include metal cutting, engraving, and welding.

### **1.6 Assumptions and Limitations**

The following are assumptions and limitation for the project.

#### **1.6.1 Assumptions**



The assumptions are as follows:

- Expected output of this project is a designed prototype and analysis of using LabVIEW for this project.
- Prototype will operate in a climate-controlled environment.
- The minimum accuracy requirement for the prototype is the minimum distance that the provided stepper motors can move.
- All required hardware and software has been provided.

### **1.6.2 Limitations**

The limitations are as follows:

- FPGA does not have trigonometric functions.
- Final prototype size is fixed at measurements of provided hardware.
- No additional power requirements than what is needed for the provided hardware.
- The physical size of the prototype will be determined by the size of the provided hardware. The product if put into production at a later time will be scalable to meet the end-user requirements.
- The system will not draw a shape that is outside the axis movement range.
- The FPGA card has a limited number of outputs.
- The group was limited by the software made available by CSG.

### **1.7 Expected End Product and Other Deliverables**

The end product will be the design of a prototype that will use a scanner to trace a shape in two dimensions. Deliverables are as follows:

- Project Plan
- Bound Project Plan
- Poster
- Design Plan
- Bound Design Plan
- Fix Broken Z-Axis (Unrealized)
- Schematic of Hardware configuration
- LabVIEW Code in 2 Dimensions
- Final Report

## **Part 2 Analysis of Hardware and Software Approach**

### **2.1 Software Approach**

#### **2.1.1 Approaches Considered**

Dividing the complete program across LabVIEW VI and the FPGA allows for better control of the system. The FPGA allows for a fixed clock rate which would not be possible within the windows environment. This is because the processing is done in parallel within the FPGA and the system will always run at a fixed rate which can be specified for the chip. Fixing the rate at a frequency which the motor controls respond to is necessary for correct operation of the system.

A custom produced chip could also do the job of the FPGA, although for design and prototype implementation this would not be practical. A new chip would have to be designed for each version of the software which would be neither cost effective nor time efficient. Allowing a PC to process the shape first significantly saves on the size of the FPGA as well as the onboard memory which is necessary. A PC will also perform at a much faster speed than the FPGA. A PC is also capable of taking input from a more varied list of peripherals as well as in a greater number of formats.

**User Level:** The LabVIEW environment will run on the computer and serve as a GUI for the user. This will allow the user to choose a shape to use and to display any output that may be needed for the user. Possible outputs will be error messages, messages signaling the state of the machine, and successful completion messages. The user interface will also contain a stop button to implement system stop at any point while the system is running automatically.

**System Level:** The LabVIEW environment will perform other actions than just to serve as a user interface. Once the user selects a shape, VI will process the shape into a collection of instructions. The instructions will then be grouped into a packet and buffered so that the FPGA can fetch them as necessary. From there, VI will signal the FPGA that packets are ready in the buffer and then wait until the FPGA returns a signal confirming the completion of the design. Once the design is finished the user will be returned to the default screen where they can choose another shape. If an error occurs at any point during operation the system will perform a system stop, return to the default screen and display the correct error message.

There are two approaches can be considered for the design of the LabVIEW VI. One design assumes that the entire list of packets can be either buffered to the FPGA or to memory on the PC. It is possible that the FPGA will only be able to read a minimal amount of data from minimal amount of inputs. An alternate design to solve this problem involves making changes to the last two states,

buffer shape and wait. LabVIEW will have to process one packet at a time and buffer it to memory instead of buffering the entire shape. The system will then wait for the FPGA to process the packet and then will return to the buffering stage to buffer the next packet. When the last packet has been sent LabVIEW will return to the default user input screen.

The software design for the FPGA can be broken down into four major states just like the LabVIEW VI. The figure below represents these four major states and shows how the design switches between these states. The FPGA switches between, waiting for new packets, fetching the available packet, processing the packet and then sending the data to the motors. The FPGA cycles through these states without stopping and requiring no user input.

The actual final design decided upon varied slightly from the approaches considered. This is because programming in LabVIEW is logically more similar to programming in a software language versus programming in an HDL language.

### 2.1.2 FPGA Design:

The software design for the FPGA can be broken down into four major states. The figure below represents these four major states and shows how the design switches between these states.

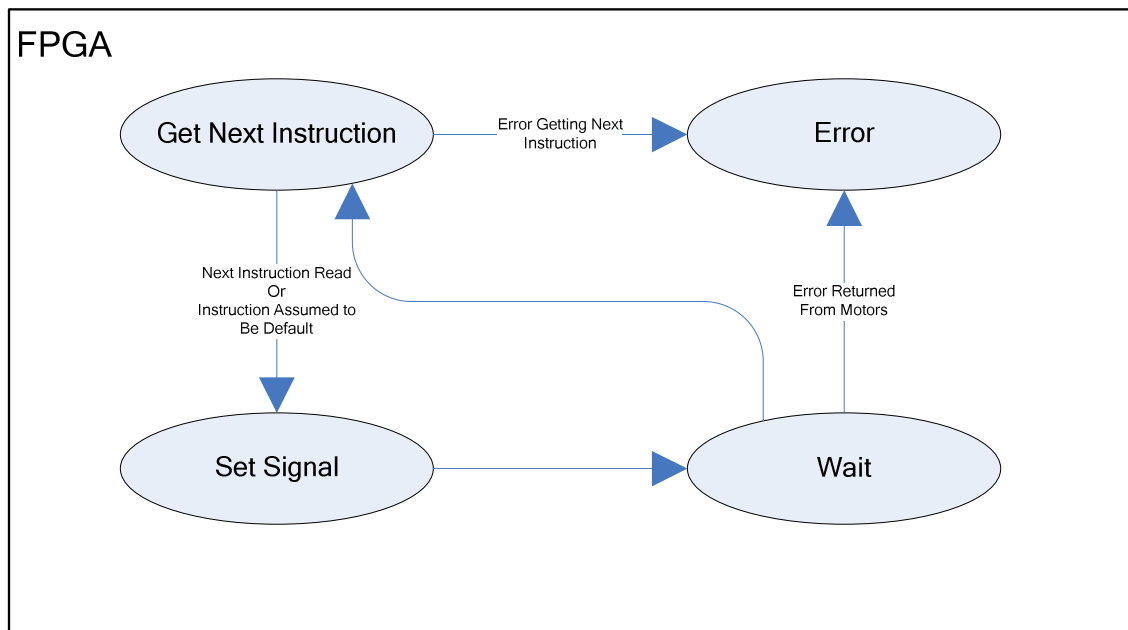


Figure 1: FPGA Software Design

**Get Next Instruction:** The default state of the FPGA. In this state the FPGA retrieves an instruction from a memory queue on the FPGA itself. If no

information is on the stack then values will be set so that the system will stay motionless. The system then moves onto the set signal state.

**Set Signal:** The FPGA will set the correct output signals coming out of the PXI and then proceed to the wait state.

**Wait:** The FPGA will wait a fixed amount of time, holding the signals long enough for them to be recognized as a valid signal. Finally the system will return to the get next instruction state.

**Error:** It is possible that an error could occur in two of the other states. If an error occurs the system will enter into dead end state and send an error message back to the VI. To FPGA will need to be restarted for the system to begin functioning again.

### 2.1.3 LabVIEW VI Design:

The design of the software for the LabVIEW VI can be broken down into four major states. The figure below represents these four major states and shows how the design switches between these states. The design follows the approach where all of the data can be placed on the FPGA at one time. This simpler design works because a stack is being used and the LabVIEW VI simply has to add the instructions to the end of a queue. Addressing the issue of running out of memory in the queue is not considered since it could be added to the VI and FPGA at a later date simply as an interrupt that stops the buffering until more memory is available.

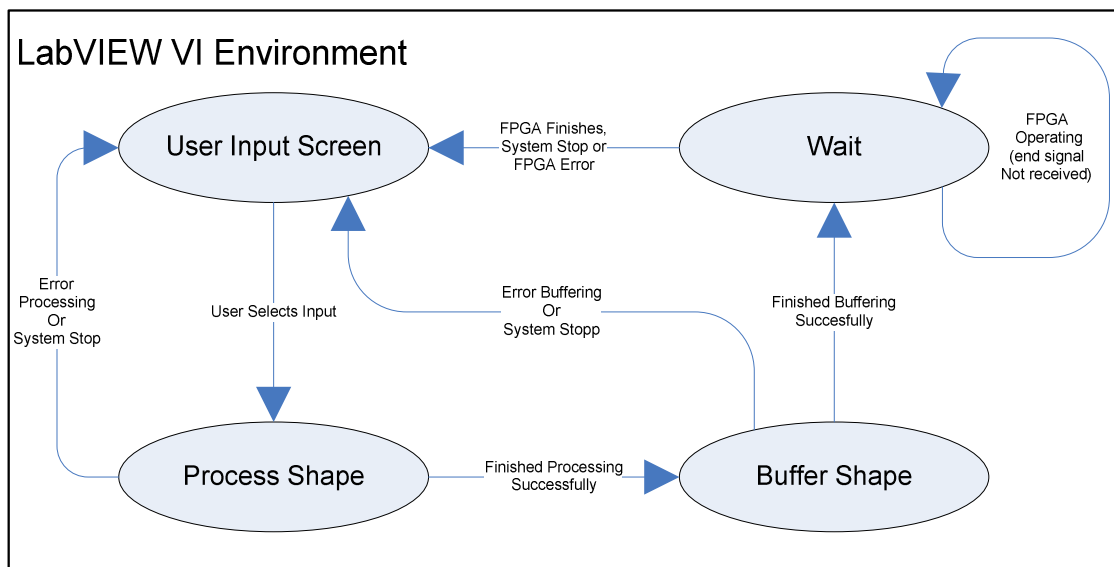


Figure 2: LabVIEW VI Software Design

**User Input Screen:** The default state of the system is the user input screen. This is where the user will select the shape that they wish to have the assembly produce. If an error occurs anywhere during any other state the system will fall back to the user input screen after displaying any necessary error messages.

**Process Shape:** Once the user has selected a shape the rest of the program is automated. The LabVIEW VI environment will process the shape and change the data into instructions. The instructions may also be preprocessed, in which case they will simply be loaded. If an error occurs the system will display the error if necessary and return to the user input screen.

**Buffer Shape:** Once the shape is separated into instructions it will be prepared for output to the FPGA by being separated into packets and stored. The shape can be buffered to one of two locations, either on to the FPGA if the memory is sufficient or on to the PC. If an error occurs the system will display a message if necessary and return to the user input screen.

**Wait:** In this state the system waits for the FPGA to send back a termination signal. This signal may be an error or it may be that the system finished successfully. The returned signal will contain any information that may be necessary to display an error. If no return code has been sent yet, the system will simply wait at this state.

#### 2.1.4 Implementation Process Description

The first piece of software to be created was two files that would be used to test the motors and their connectivity to the PXI. These files were created to simply send preset values to the motors for a fixed amount of time. This would allow tests to be done later to see how long signals had to be held high, how rapidly the next movement could be sent and how much the system would shake.

The next step was to create the LabVIEW code for the FPGA. By setting static values in the FPGA and making a temporary front end it would be possible to test the FPGA code without the more dynamic code of the LabVIEW VI interacting with it.

The FPGA was fed a static array which could be filled with data from the temporary front end. Three of the four stages for the FPGA lied within a sequential unit. The first unit in the sequence was the 'get next instruction' state, the next was the 'set signal' state and finally the 'wait' state. The 'wait' state also was chosen to serve as the point in which the values in the array would be shifted since the longest time would be spent in that state. The shifting could also have occurred in the 'set signal' state however if it was found to take additional time. The fourth state, error, was not included because it would work

in conjunction with and was heavily dependent upon the separate LabVIEW VI that would be implemented at a later time

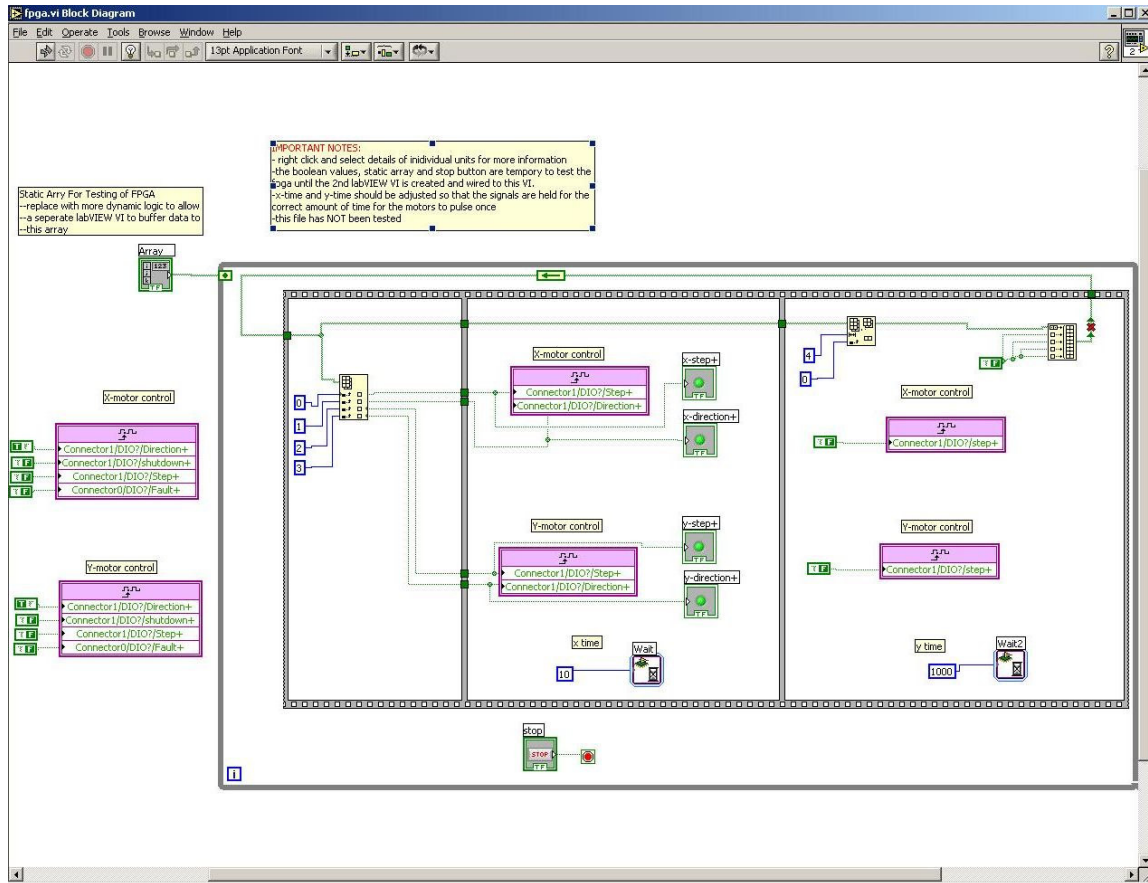


Figure 3: *FPGA code static values and front end control*

The front end also had LED's to display which signals were set high for the motors to evaluate if that value was being set correctly. Finally a stop button was added in case difficulties occurred during the testing. If the system was stopped the code would need to be recompiled to the FPGA to start it again. While this seemed troublesome at first, if the system would be stopped, then it was likely that the either the code or the static values needed to be changed which would require a new compile regardless.

Lastly the LabVIEW VI code will be created and will be used to replace the temporary static array values in the FPGA code as well as its front end. While this design will originally have one fixed shape to select, it will later be possible to select more shapes or possibly even to load a shape from a separate file.

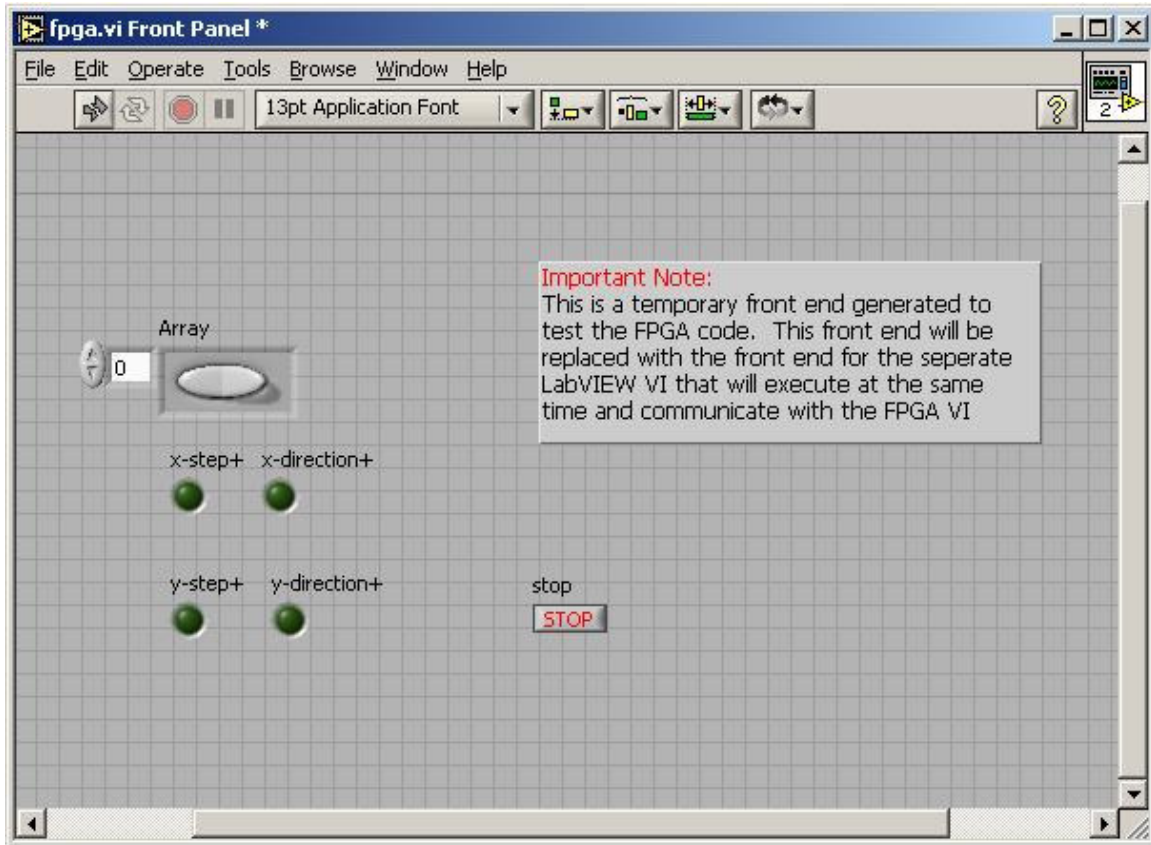


Figure 4: *Temporary front end for the FPGA*

### 2.1.5 Testing of the End Product Software and Its Results

To test the motors two simple LabVIEW files will be created to simply test sending a signal to the motors. These two test programs set outputs to pre-set values for a set amount of time. Changing the static values in the test files would allow timings and delays to be tested.

### 2.1.6 End Results of the Project Software

The LabVIEW VI plus the FPGA where capable of everything we wanted to do plus much more. While the FPGA lacked some of the functions of the LabVIEW VI, they can be used jointly to overcome these obstacles. LabVIEW appears difficult to learn at first, but is surprisingly easy if the user has experience with C and HDL. LabVIEW allows for the serial logic of software languages as well as the parallel functionality of HDL. The LabVIEW program has extensive help and tutorial that assist when creating programs. The help menus also allow for the searching and insertion of functions once they are located.

## **2.2.The Hardware Approach**

The hardware approach and result are as follows:

### **2.2.1 Functional Requirements for the End-Product**

The end product of the hardware should be able to:

- The stepper motor driver should receive input signals from the FPGA to drive the scanner.
- The signal sent to the stepper motor driver should be the right pulse size to move the stepper motor one step per pulse.
- The scanner should draw shapes which meet the scanner dimension constraint.
- The scanner should be able to retrace the shape it draws previously with minimum error as possible.

The end product of the hardware should not:

- The scanner should not wobble too much in the instant the axis stop moving.
- The scanner will not be able to move less than the minimum step the motor can move.

### **2.2.2 Resultant Design Constraints**

The design constraints of the hardware system are listed bellow:

- The scanner will not draw a shape which is out of the scanner movement range.
- The scanner will only draw shapes predetermined by software only.
- The hardware will operate at normal room humidity and temperature.
- The system's movement accuracy depends on the minimum distance the stepping motor can move per step.

### **2.2.3 Approaches Consideration**

Some approaches the team considered:

Controlling the stepper motor using analog controller

- Easier to configure compare to digital controller.
- Inherited from previous group
- Harder to connect to the PXI with the FPGA.
- The controller is not a National Instrument device.
- This was not the purpose of the project.

Controlling the stepper motor using digital stepper motor driver

- Easier to connect to National Instrument SCB-68 and PXI with FPGA.
- Inherited from previous group.



Figure 5: Stepper Motor Driver



- No manual obtained from previous group or from the internet.

We used the second approach, because:

- The initial project was to use National Instrument product to complete the system.
- The easiness of connecting the FPGA with the SCB-68 board
- The easiness of controlling digital signal using LabVIEW compare to analog signal.

## 2.2.4 Detailed Design

The Parker scanner has the ability to move in three dimensions. However, the limiter on the z-axis of the system is broken. One of the tasks is to replace the limiter so the scanner can move in three dimensions. The end product will use a laser pointer attached to the lever on the z-axis to trace the shape on the desired medium.

Figure 7 is a diagram of the SCB-68 board that is used to bridge the stepping motor and the FPGA. The cables from the screw terminals will be

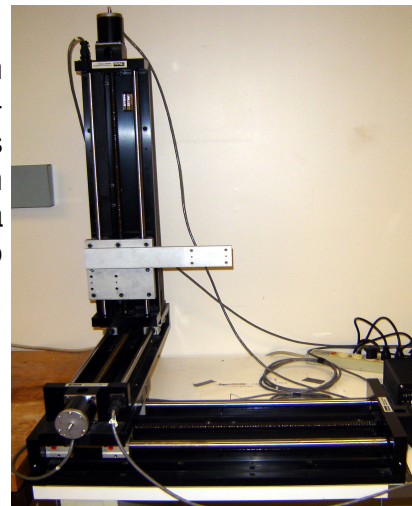


Figure 6: Scanner

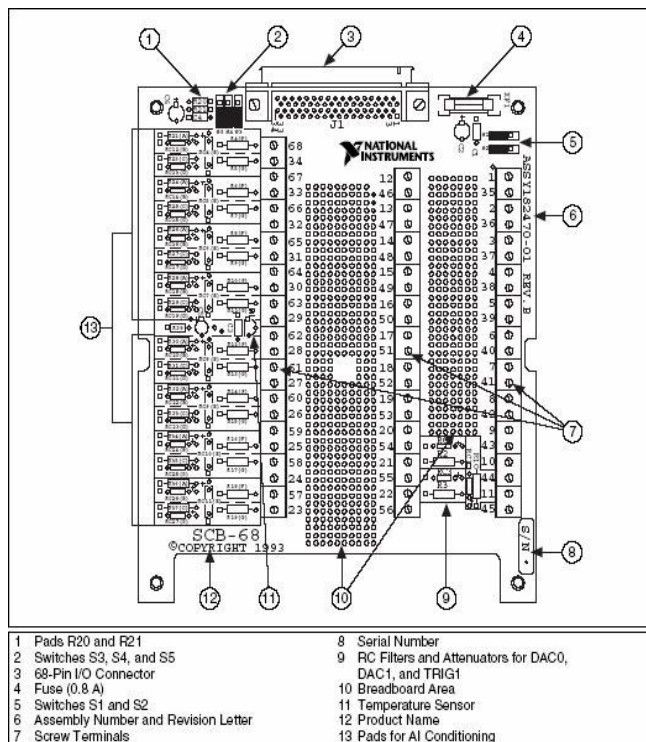


Figure 7: SCB-68 Board

connected directly to the 25-pin D-type port connector. The SCB-68 will get the instruction from the FPGA and send the step direction and the shutdown input signals to the stepping motor. There will be two SCB-68 boards used for the system. One board will drive two stepping motor drives and the other one will drive the third stepping motor.

Figure 8 is a diagram of the PK3 stepping motor drive which gets the input from the SCB-68. All control signal connections are made to a 25-pin D-type

connector. The system will primarily use six pins as the main input for the driver. A pulse input on pin 1 and 14 will step up or step down the motor respectively. The input to pin 2 or 15 will change the direction the motor moves. Pins 16 and 17 are used in conjunction with dipswitch 2 of the driver to either energize or de-energize the motor.

The motor is used for the system is a Compumotor QM motor type, the outputs are connected in parallel. This is shown in Figure 9.

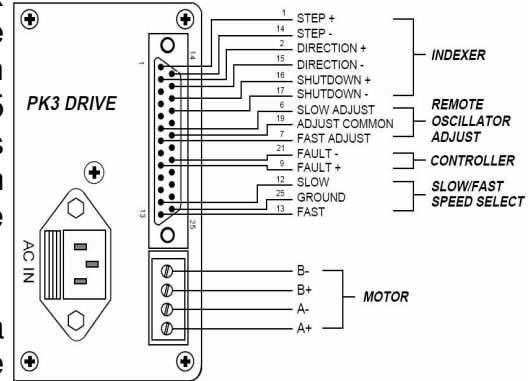


Figure 8: PK3 Drive Pins

MAKE	TYPE	A+	A-	B-	B+	NOTES
Digiplan/Compumotor QM Motor	8-lead	Red & Blue	Blk & Yellow	Wh & Brn	Green & Org.	

Figure 9: QM Motor Wiring

The NI PXI as shown in **Figure 8** sends program instructions to the FPGA. The FPGA then sends step, direction and shutdown signals through the SCB-68 box to the PK3 motor drivers. The PK3 motor drivers operate the motors as instructed. The motors can send back an error message to the FPGA.



Figure 10: FPGA Card

The NI PXI-1002 chassis, shown in Figure 11 is used to host the NI PXI-8176. The NI PXI controller has a 1.26GHz Pentium III processor. The NI PXI-7831R is the FPGA that controls the axis movement. The core of the NI PXI-7831R right above is an FPGA which is reconfigurable with the LabVIEW FPGA module. It will get instructions from LabVIEW and output the signals to the SCB-68.

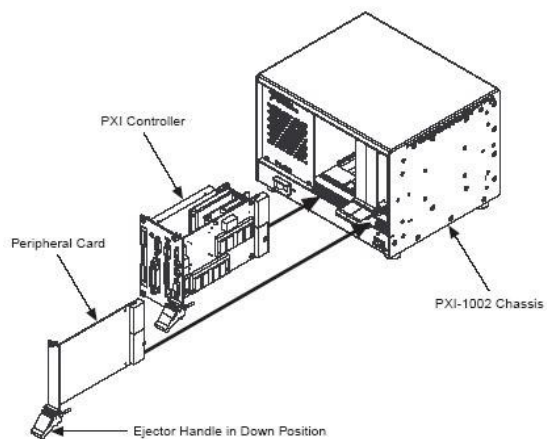


Figure 11: NI PXI

The system flow of the prototype is shown in the figure 12.

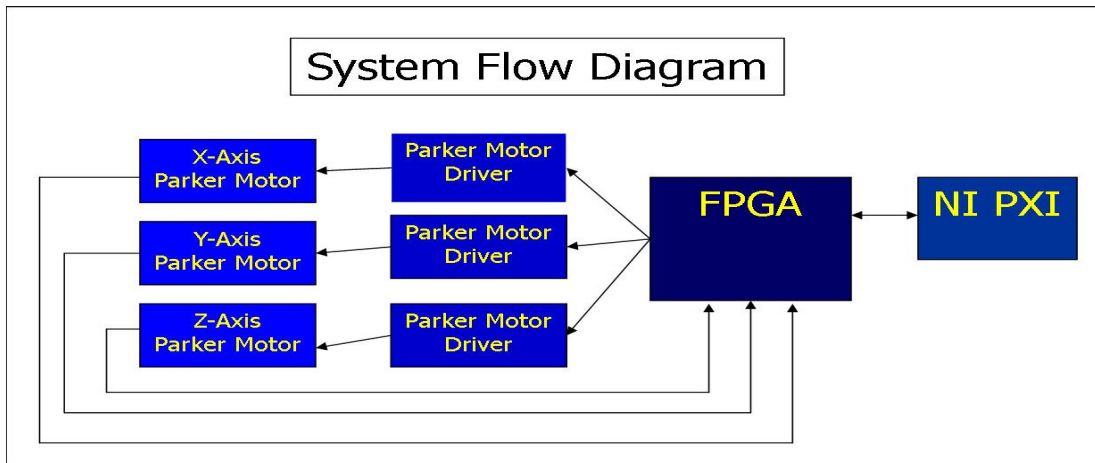


Figure 12: System Flow Diagram

### 2.2.5 Implementation Process Description

Most of the hardware for the project was inherited from another group who works on this project previously. However, there was not enough documentation or hardware manuals obtained from the previous group. In fact, only one hardware manual was obtained from the previous group, while the rest of the hardware manuals were obtained by searching from the hardware companies' websites. This problem could have been countered by the first group having a thorough documentation.

In addition, the scanner obtained for the project was an early 1980's product which was donated to the previous team by their advisor. Hence, our team could not find any information on its manual. Moreover, due to this factor of extreme age, the team could not find the limiter of the Z axis that was broken by the previous group. As a result, the hardware implementation process of the project was slow than would have been ideal. Having a newer version of a scanner might have significantly helped to solve this problem of elusive documentation.

Another problem the team experienced because of the vague information that was acquired from the hardware manuals was the lack of a complete wiring diagram for the scanner. The wiring process then took longer than the team anticipated. Furthermore, due to the software problem, the team cannot test the system wiring as well as the system functionality.

Wire Color	Function	SCB-68 Pin#	
		X-axis	Y-axis
Blue	Step +	46	60
Grey	Step -	12	26
Orange	Direction +	47	59
Brown	Direction -	13	25
Red	Shutdown +	48	58
Green	Shutdown -	14	24
Black	Fault +	49	57
White	Fault -	15	23

Table 1: Wiring to the SCB-68

### **2.2.6 End-Product Testing Description**

The testing that has been conducted so far was only to check the connection between the stepper motor attached to the scanner and the stepper motor driver. This testing was done by the team to test if the scanner is working properly. This test was a success, and the scanner and stepper motor drivers have been shown to be functional.

As of now, the team has not been able to attempt any controlled testing on the system due to the unavailability of the LabVIEW software to recognize the FPGA. This leads to the inability to send any user controlled signal to the stepper motor driver. Hence, the team cannot test user control of the system.

### **2.2.7 End Result of the Project Hardware.**

The hardware is connected already. However, the team cannot be sure of the wiring connection from the scanner bulk because there are no manuals and descriptions found regarding the function of the particular cable. In addition to that, the team cannot extensively test the wiring between the hardware since there is software problem in recognizing the FPGA.

## **Part 3 Resources and Schedules**

### **3.1 Personnel effort**

#### **3.1.1 Original Personnel Hours**

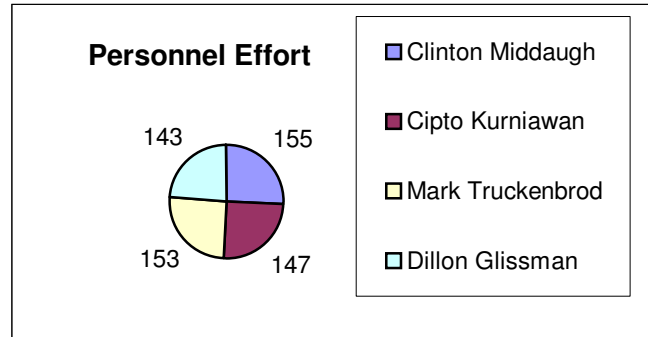


Figure 13: Original Personnel Hours

#### **3.1.2 Revised Personnel Hours**

Task	Cipto	Clinton	Dillon	Mark	Total
Project Definition	5	3	4	4.5	16.5
Project Poster	12	2	8	11	33
Explore Technology Limitations with FPGA	1	6	1	0.5	8.5
Design Prototype and Purchase Hardware	15	10	2	5	32
Assemble the Prototype	15	21	5	26	67
Software Implementation in Two Dimensions	10	19	39	10	78
Prototype Testing and Debugging in Two Dimensions	15	18	18	19	70
End Product Documentation	10	13	10	15	48
End Product Demonstration	15	10	10	10	45
Continuing Project Evaluation	49	53	46	52	200
Total	147	155	143	153	598

Table 2: Revised Personnel Hours

### 3.1.3 Actual Personnel Hours (to date with one month left)

Task	Cipto	Clinton	Dillon	Mark	Total
Project Definition	5	3	4	4.5	16.5
Project Poster	12	2	8	11	33
Explore Technology Limitations with FPGA	1	6	1	0.5	8.5
Design Prototype and Purchase Hardware	1	0	0	1	2
Assemble the Prototype	5	0	0	2	7
Software Implementation in Two Dimensions	2	5	15	3	25
Prototype Testing and Debugging in Two Dimensions	1	0	1	1	3
End Product Documentation	3	2	3	5	13
End Product Demonstration	0	0	0	0	0
Continuing Project Evaluation	31	66	47	57	201
Total	61	84	79	85	309

Table 3: Actual Personnel Hours

## 3.2 Financial Resources

### 3.2.1 Initial Financial Resources Amounts

#### Financial Requirements

The client has supplied most parts. Other parts are estimated to cost as follows:

Project Poster	\$65
Laser Pointer	\$50
Limiter for step motor	\$50
Used computer monitor	\$50
Binding and printing	\$10
Estimated labor costs	\$6428.50
Total Estimated cost	\$6703.50

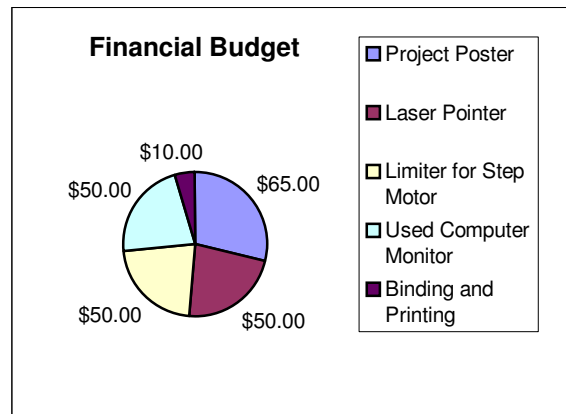


Figure 14: Initial Budget

### 3.2.2 Revised Projected Financial Resources Amounts

Item	Cost
Project Poster	\$74.50
Laser Pointer	\$50
Limiter for step motor	\$50
Used computer monitor	\$50
Wire	\$50
Electronics Parts	\$25
Binding and printing	\$10
Estimated labor costs	\$6,428.50
Total Estimated cost	\$6,738.00

Table 4: Revised Budget

### 3.2.3 Actual Financial Resources Amounts

Project Poster	\$74.50
Laser Pointer	\$0
Limiter for step motor	\$0
Used computer monitor	\$50
Wire	\$0
Electronics Parts	\$0
Binding and printing	\$10
Estimated labor costs	\$3321.75
Total Estimated cost	\$3456.25

Table 5: Actual Budget

## 3.3 Schedules

### 3.3.1 Initial Gant Chart of Tasks

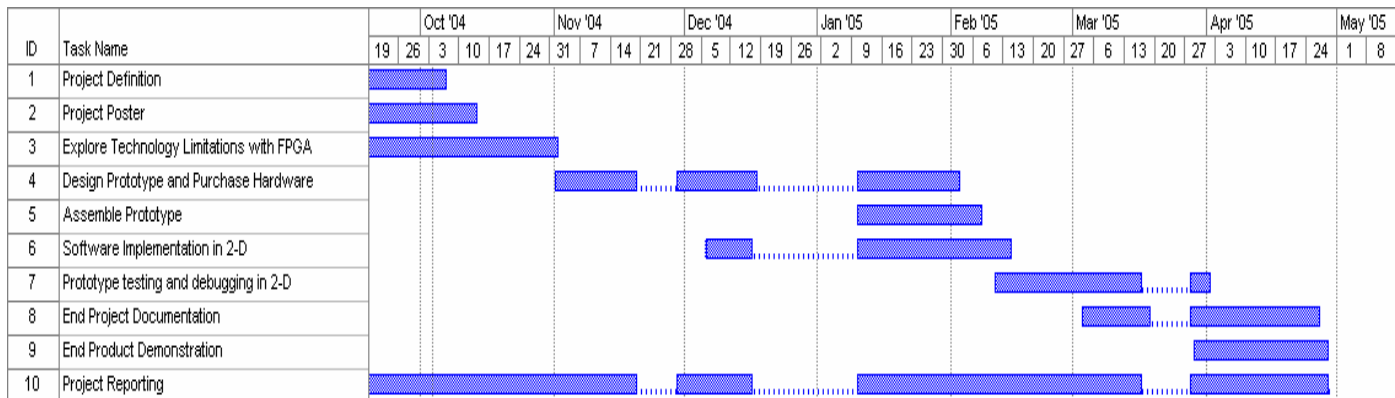


Figure 15: Initial Gant Chart



### 3.3.2 Revised Gant Chart (No revisions had been necessary)

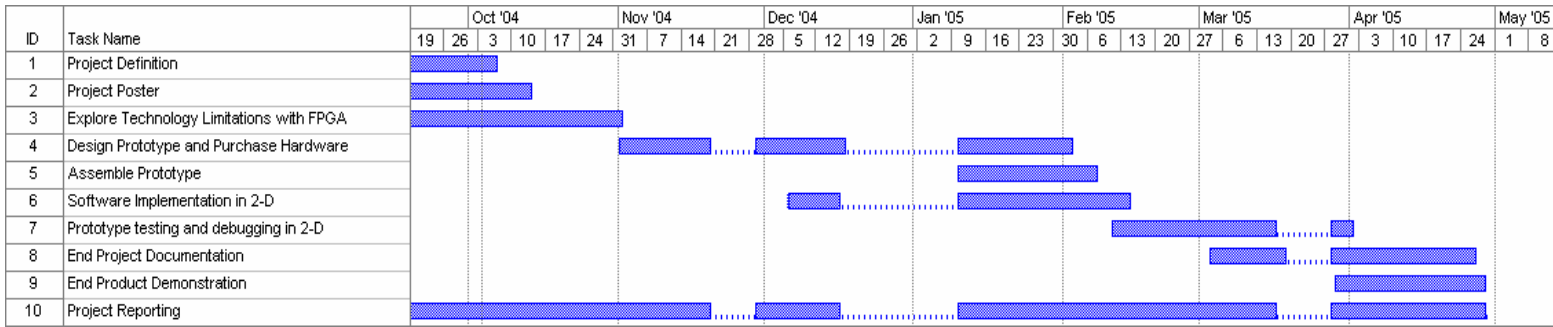


Figure 16: Revised Gant Chart

### 3.3.3 Actual Gant chart of tasks as they have been completed

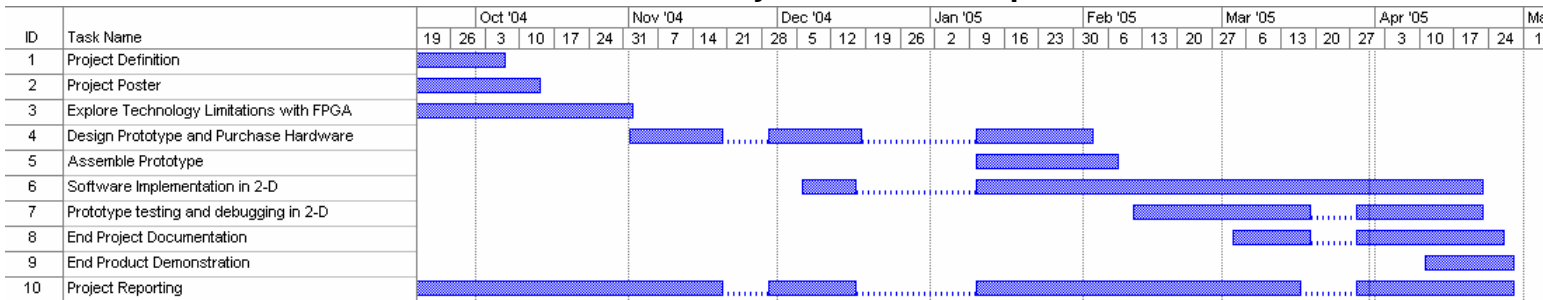


Figure 17: Actual Gant Chart

### 3.3.4 Explanation of Differences between revised and actual Gant charts

The differences between the two revised and actual Gant charts can be explained by the problem the project has run into with the LabVIEW VI program. Thus far the group has been unable to find a way to get the LabVIEW program to recognize the FPGA Card as an execution Target. The PXI computer shows the FPGA card as recognized and fully functional, so some sort of module interaction problem might be what is causing the issue. Until the problem is resolved, testing, and debugging will not be possible.



## **Part 4 Closure Materials**

The following are Closing Materials.

### **4.1 Product Evaluation**

The following lists the deliverables for the project, and whether the deliverables have been greatly exceeded, exceeded, fully met, partially met, not met, or not attempted:

ID	Task Name	Aug 22, '04	Sep 12, '04	Oct 3, '04	Oct 24, '04	Nov 14, '04	Dec 5, '04	Dec 26, '04	Jan 16, '05	Feb 6, '05	Feb 27, '05	Mar 20, '05	Apr 10, '05	May 1, '05
1	Project Plan													
2	Bound Project Plan													
3	Project Poster													
4	Design Plan													
5	Bound Design Plan													
6	Schematic of Working Hardware Configuration													
7	Labview Code Implemented in 2 dimensions													
8	Output of Working Hardware/Software System													
9	Unbound Final Report													
10	Bound Final Report													

Figure 18: Deliverables Schedule

- Project Plan: Fully Met
- Bound Project Plan: Fully Met
- Poster: Fully Met
- Design Plan: Fully Met
- Bound Design Plan: Fully Met
- Schematic of Hardware configuration: Fully Met
- LabVIEW Code in 2 Dimensions: Partially Met
- Output of Working Hardware/Software System: Not Met
- Unbound Final Report: Fully Met
- Bound Final Report: Not Met

### **4.2 Commercialization**

This project is already widely available on the public market and would be impractical to be commercialized by the group because of the high cost of the scanner and the LabVIEW software package.

### **4.3 Recommendations for Additional Work**

It is recommended that the project be discontinued. The LabVIEW software is hard to set up, and is unreasonable to think a Senior Design team will be able to set it up in the limited time available. The scanner and stepper motors are old and documentation is scarce.

#### **4.4 Lessons Learned**

##### **4.4.1 What Went Well**

- Group Communications
- Coding in LabVIEW
- Scanner is functional in two dimensions

##### **4.4.2 What did not go well**

- Slow start, project not well defined in the beginning
- Misleading information was provided by previous group (Dec 03-07), making initial progress slow
- LabVIEW module interaction with FPGA
- Correct software was not installed on PXI

##### **4.4.3 Technical Knowledge Gained**

- Use of LabVIEW and FPGA
- Scanner and stepper motor operation
- Use of PowerPoint for poster design and presentation
- Wiring of hardware

##### **4.4.4 Non-technical Knowledge Gained**

- Communication
- Group organization
- Aggressiveness in getting projects started

##### **4.4.5 What we would do differently if we did it again**

- Start bulk of work earlier
- Could have attempted to find a newer scanner to use

#### **4.5 Risk and Risk Management**

##### **4.5.1 Risks**

- Loss of team member

- Stepper motors not working
- Devices not communicating correctly

#### **4.5.2 Risk Management**

- Thorough documentation
- Tested motors as soon as possible to make sure they work

## **Part 5: Contact Information**

### **5.1 Client Information**

Name: Dr. Mani Mina  
Address: 341 Durham  
Ames, IA 50011  
Phone Number: (515) 294-3918  
Email: mmina@iastate.edu

### **5.2 Faculty Advisor Information**

Name: Dr. Mani Mina  
Address: 341 Durham  
Ames, IA 50011  
Phone Number: (515) 294-3918  
Email: mmina@iastate.edu

### **5.3 Project Team Information**

Name: Dillon Glissman  
Address: 2111 Frederiksen Ct  
Ames, IA 50010  
Phone Number: (515) 572-7776  
Email: dgliss@iastate.edu

Name: Cipto Kurniawan  
Address: 218 Walnut Ave #1  
Ames, IA 50010  
Phone Number: (515) 441-0198  
Email: cipto@iastate.edu

Name: Clinton Middaugh  
Address: 4316 Maricopa Dr #6  
Ames, IA 50014  
Phone Number: (515) 480-0367  
Email: middaugh@iastate.edu

Name: Mark Truckenbrod  
Address: 1305 Georgia Ave  
Ames, IA 50014  
Phone Number: (515) 231-1977  
Email: mkt@iastate.edu

#### **5.4 Closing Summary**

In conclusion, the problems of precise and steady scanner movement, with user input tracing are thorny issues. Given the National instruments and Parker hardware and LabVIEW software provided, the team went about designing a solution to the problem. While some software issues have kept the solution from full implementation, there is a good chance the software will be made to work and the versatility of LabVIEW and National Instruments hardware will be displayed.

## **Part 6: References**

NI 7831 R FPGA:

[http://www.ni.com/pdf/products/us/04\\_3632\\_301\\_101.pdf](http://www.ni.com/pdf/products/us/04_3632_301_101.pdf)

National Instruments PXI 1002:

<http://www.ni.com/pdf/manuals/322650a.pdf>

National Instruments SCB-68:

<http://www.ni.com/pdf/manuals/320745b.pdf>

Parker PK3 Stepper Driver Manual:

<http://www.compumotor.com/manuals/pk3.pdf>

Senior Design Project Dec 03-07 Documents:

[www.seniord.ee.iastate.edu/dec0307](http://www.seniord.ee.iastate.edu/dec0307)

## **Part 7 Appendices**