FPGA BASED FLIGHT CONTROLLER

Project Report

Submitted in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY
IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

Athul John Kurian (MDL17EC030)
Medha Lakshman Rao (MDL17EC078)
Omar Bin Shafi (MDL17EC091)
Sebastian James (MDL17EC109)

Department of Electronics Engineering
Govt. Model Engineering College
Thrikkakara, Cochin - 682 021
APJ Abdul Kalam Technological University

December 2020

FPGA BASED FLIGHT CONTROLLER

PROJECT REPORT

Submitted to
the APJ Abdul Kalam Technological University
in partial fulfillment of the requirement for the award of Degree of
Bachelor of Technology in Electronics and Communication Engineering

BY
ATHUL JOHN KURIAN (MDL17EC030)
MEDHA LAKSHMAN RAO (MDL17EC078)
OMAR BIN SHAFI (MDL17EC091)
SEBASTIAN JAMES (MDL17EC109)

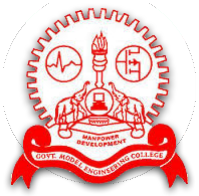
Department of Electronics Engineering,
Model Engineering College,
Thrikkakara, Kochi-682021,
Kerala.

DECEMBER 2020

DEPARTMENT OF ELECTRONICS ENGINEERING

MODEL ENGINEERING COLLEGE

THRIKKAKARA, KOCHI-682021



CERTIFICATE

This is to certify that this Project entitled ***FPGA BASED FLIGHT CONTROLLER*** is the bonafide record of work carried out by **ATHUL JOHN KURIAN (MDL17EC030), MEDHA LAKSHMAN RAO (MDL17EC078), OMAR BIN SHAFI (MDL17EC091), SEBASTIAN JAMES (MDL17EC109)** in partial fulfillment of the requirements for the completion of Degree of Bachelor of Technology in Electronics and Communication Engineering, at the Department of Electronics Engineering, Model Engineering College, Thrikkakara, Kochi.

Mr. Jagadeesh Kumar P
Project Guide

Ms. Thushara H P
Project Co-ordinator

Dr. Jayasree V K
Head of the Department

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Almighty for helping us to make this possible and complete this work on time. We would like to express our sincere gratitude to everyone who assisted us in completing this project. We express our gratitude towards Prof. (Dr.) Vinu Thomas, Principal, Model Engineering College and Prof. (Dr.) Jayasree V.K, HOD of Electronics Engineering, Model Engineering College. We express our sincere gratitude towards our coordinator Ms. Thushara H P, Assistant Professor, Department of Electronics Engineering, for giving us permission to commence this project. Special thanks to our guide Associate Prof. Mr. Jagadeesh Kumar P, Department of Electronics Engineering, whose guidance and encouragement helped us throughout this endeavour. Last but not the least we would like to thank all our fellow classmates who gave us constant support for completing this venture successfully.

ABSTRACT

Flight control system (FCS) is the brain of the any aircraft especially in the Unmanned Aerial Vehicles (UAVs). FCS can use various controllers such as conventional Proportional Integral Derivative (PID) Controller, combination of PID controller with Kalman filter, fuzzy logic controller and Field Programmable Gate Array (FPGA) / Digital Signal Processor (DSP) based controller, etc. This project presents the development of a UAV flight controller using a complete parallelism embedded system as an FPGA. Many solutions for UAV´s flight controllers are based on embedded sequential systems. This project presents the development of the FCS for small UAVs namely, purpose of work, flow diagram, advantages and disadvantages. FPGA/DSP based small UAVs are better one, as it is a low power, fast response, less volume and compatible to numerous applications. The aim of this project is to achieve a set of suitable characteristics compatible with applications in the field of unmanned flight control.

Contents

List of Tables

List of Figures

ABBREVIATIONS

BLDC Brushless Direct Current

ESC Electronic Speed Controller

FCS Flight Control System

FPGA Field Programmable Gate Array

FSM Finite State Machine

HDL Hardware Description Language

I2C Inter Integrated Circuit

IDE Integrated Development Environment

IMU Inertial Measurement Unit

ISE Integrated Synthesis Environment

ML Machine Learning

MVP Minimum Viable Product

PID Proportional Integral Derivative

PPM Pulse Position Modulation

PWM Pulse Width Modulation

RC Radio Control

RF Radio Frequency

SPI Serial Peripheral Interface

UAV Unmmaned Aerial Vehicle

VHDL Very High Speed Integrated Circuit Hardware Description Language

Chapter 1

INTRODUCTION

The UAV's (Unmanned Aerial Vehicle) or drones as they are commonly known, are aircrafts which can be controlled remotely or flown autonomously. These devices have become very popular in recent years and can be used for several applications. Many solutions are based in embedded sequential systems, however the more processes these systems execute the greater impact it will have on variables such as precision, speed of response and synchronism. This article presents a solution for the UAV's Flight Controller integrated in an embedded concurrent system such as an FPGA. The FPGA provides enough power of calculation to extend the intelligence of the system with more autonomy, excellent synchronism and improved response times.The goal of the Flight Controller System is to help the user with automatic compensation in all the maneuvers of the aircraft in case of any disturbance. Field Programmable Gate Array (FPGA) provides parallel processing (all the bit at a time) and having large numbers of gates on a single chip (essential for less volume). FPGA also provide cost savings platform. The software of the systems would also have a huge impact on the overall design, for which code parallelization provides faster speed in FPGA and other hardware components.The inputs of the system are the signals delivered by the RF receiver which delivers the commands coming from the user and the sensor's signals which give the direction and position of the aircraft in real time.

1.1 Aim

1. Main idea: To design a flight controller (FC) system on an FPGA board using Verilog to program the FPGA.
2. To deploy the code in Xilinx Virtex - 6 FPGA development board to tune the flight parameters to achieve stable flight using a radio transmitter/similar manual input.

1.2 Scope and Relevance

1. Can be used for UAV flight stability
2. Faster response of flight controller - parallel processing capability of FPGA.
3. From entertainment to military purposes, drones are used in various applications and the scope of drone technology is limitless.
4. Lots of scope for FPGA-based flight controllers considering the innumerable application of the drones.

Chapter 2

LITERATURE SURVEY

The insights that we have gained from referring the list of papers have been mentioned here after in this chapter.

SL NO	YEAR	TITLE & SOURCE	OBJECTIVES	METHODOLOGY	RESULT	LIMITATIONS
1	2016	**TITLE** Design, development and implementation of a UAV flight controller based on a state machine approach using a FPGA embedded system. **SOURCE** DASC	Fast Real time processing of inputs Sensor Interfacing With FPGA Combining Various Sensor inputs	Parallel processing and having large numbers of gates on a single chip And also provides a cost saving platform. Sensors measure the air vehicle's altitude and other functions and compare it to the desired states and the error signal is eliminated	Parallel Processing	Noise From Sensors Affect stability

Table 2.1: Summary of Reference Paper 1[1]

Table 2.1 summarizes the idea we have got from the reference paper. The paper provided, gave us an understanding of the range of applications drones are being used for, parallel processing capability of FPGAs enabling integration of various sensor units with the FPGA resulting in fast real time processing of inputs, requirement of PWM driver for ESC.

SL NO	YEAR	TITLE & SOURCE	OBJECTIVES	METHODOLOGY	RESULT	LIMITATIONS
2	2016	**TITLE** An analytical review on FPGA based autonomous flight control system for small UAVs **SOURCE** ICEEOT	To smoothen out MPU 6050 Output using filters. To generate PWM motor Signals corresponding to roll and pitch from the sensors	Kalman Filter and PID Controller offers short transition, good stability, anti-disturbance, good control and fulfills the requirement of real-time and accurate control. FCS The goal of the FCS is to help the user with automatic compensation in all the maneuvers of the aircraft in case of any disturbance.	Timing and Control of FPGA input signals from sensors. Synchronization of signals. Generation of output PWM signals for controlling the motors	PID - limited stability

Table 2.2: Summary of Reference Paper 2[2]

Table 2.2 gives a summary of the knowledge we have gained from the paper. We understood that the output of MPU is to be smoothened by means of filters, the equation necessary to generate PWM in terms of user input values, an overview of the block of implementing a flight controller, details regarding PID controller, combination of PID and Kalman filter, use of fuzzy based controller, timing and control of input signals from sensors.

SL NO	YEAR	TITLE & SOURCE	OBJECTIVES	METHODOLOGY	RESULT	LIMITATIONS
3	2018	**TITLE** Design and Implementation of FPGA Based Quadcopter **SOURCE** IJETSR	PWM encoder/decoder Design Study Basic Block diagram of the FC Mapping unit to produce output for motors from roll,pitch,yaw,throttle	FPGA Block Diagram PWM encoder PWM signal generator PWM decoder Mapping unit	Relation between the Motor output, Throttle Roll, pitch and yaw required.	No PPM encoder/decoder is implemented.

Table 2.3: Summary of Reference Paper 3[3]

From Table 2.3 we got an idea on how to design PWM encoder and decoder, further insights into the basic block diagram of FC, mapping output of FPGA to user defined terms for motor control.

Chapter 3

MOTIVATION AND OBJECTIVES

3.1 Motivation & Objectives

UAVs have become very popular in recent years and can be used for several applications. A flight controller is the core of any UAV and its feasibility depends on the efficiency of the flight controller. The motive is to build a well effective and efficient flight controller system.

The main objectives are:

- Testing and interfacing MPU-6050 (I2C) with FPGA board.
- Implementing PID loop and PWM encoder in the FPGA.
- Direct the input of ESC to control RPM of each motor in response to input into PID from MPU and RC transmitter

3.2 Novelty

- Different implementations other than PID
  - Kalman filter
  - Finite State Machine
  - Fuzzy based
- Convert to a minimum viable product (MVP).

- Integration of more sensors like barometer, ultrasonic to improve stability.
- Implementation of the project in VHDL / SystemVerilog as well, apart from Verilog, to portrait the benefits.

Chapter 4

BLOCK DIAGRAM

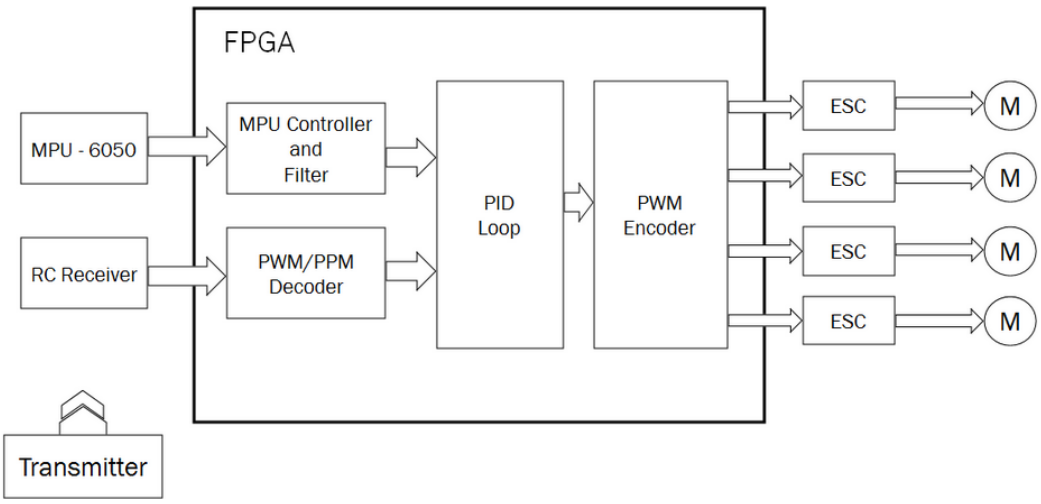


Figure 4.1: Block diagram: Flight Controller on FPGA

From Figure 4.1 we can understand that the Control of Flight controller begins with the radio receiver, which captures stick values from the radio transmitter used to control the drone. These values are interpreted as "intended" angular velocities for the drone about the three coordinate axes. Inside the FC, a control loop compares the intended velocities to the actual velocities reported by an on-board gyroscope module. Finally, it calculates what motor power setting would most quickly correct the error and communicates the result to the motors.

4.1 Methodology

4.1.1 Pulse Position Modulation

Pulse Position Modulation (PPM) is an analog modulating scheme in which the amplitude and width of the pulses are kept constant, while the position of each pulse, with reference to the position of a reference pulse varies according to the instantaneous sampled value of the message signal.

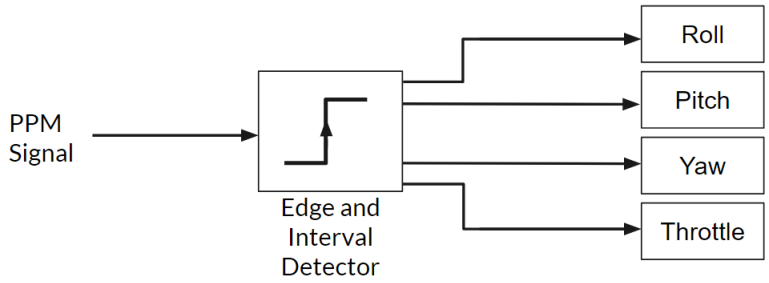


Figure 4.2: PPM - Logical Dataflow

Figure 4.2 illustrates the overview of what PPM decoder is meant to achieve. The encoded PPM which is input to the PPM decoder has an edge and interval detector which gives the user inputs as output.

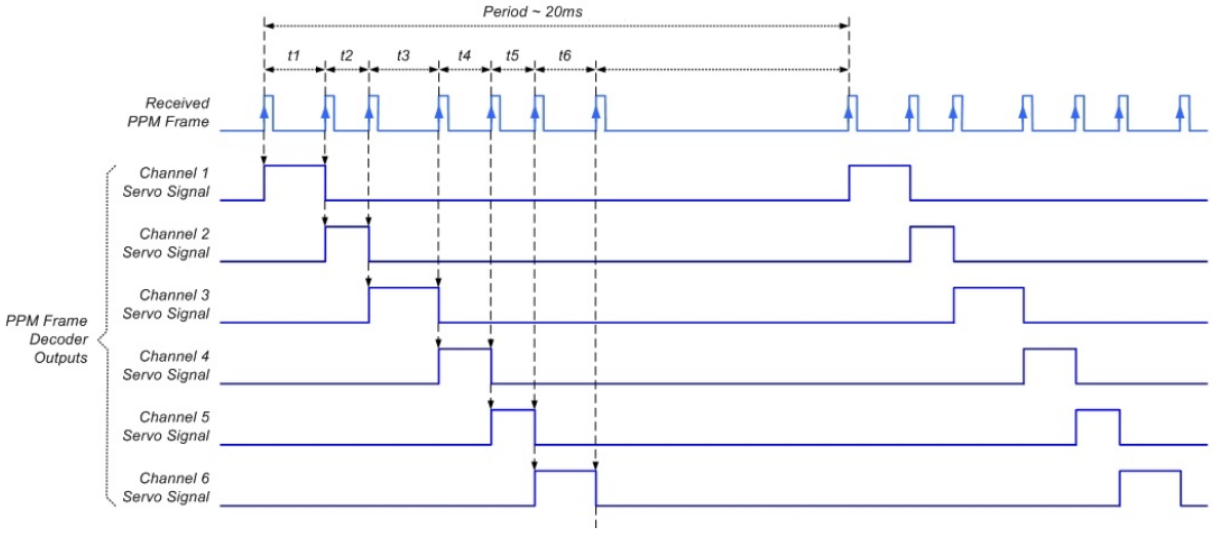


Figure 4.3: Pulse Position Modulation Decoder Dataflow

From Figure 4.3 we can observe that each pulse of the PPM signifies each channel for flight movement control which includes roll, pitch,

yaw and throttle control signals from the RC. In a PPM decoder the time interval between each pulse is measured and is manipulated to the appropriate channels in the output. In an FPGA, this can be done by using a counter to measure the length of signals between two pulses and added with pulse length of the preceding pulse to obtain each channel signals. This is the value that is set by the user. This signal is given into the PID loop for comparing this user set value with the current value from the sensor and producing a desired result signal to run the motor.

4.1.2 Pulse Width Modulation

Pulse Width Modulation (PWM) or Pulse Duration Modulation (PDM) or Pulse Time Modulation (PTM) is an analog modulating scheme in which the duration or width or time of the pulse carrier varies proportional to the instantaneous amplitude of the message signal. The width of the pulse varies in this method, but the amplitude of the signal remains constant.

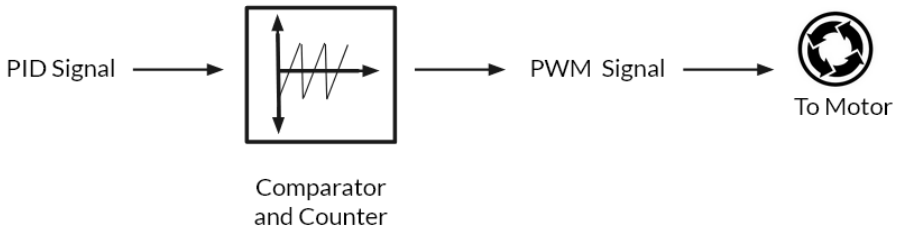


Figure 4.4: PWM - Logical Dataflow

Implementation of the Pulse Width Modulation can be done using the following logic: The PPM Decoder is used to interpret the pulse position modulation (PPM) signal provided by the radio receiver.

From Figure 4.4 we can see that a free-running counter (one that is continuously incremented) can be used to set an output depending on the counter's state. A comparator is used to compare a set threshold to the counter's current value, we can output 1 and 0 otherwise(or vice versa).

This signal is decoded by setting a counter-timer and measuring the

intervals between rising edges of the PPM signal. These intervals are then manipulated and written out to the appropriate channels in the module's output, which are determined using a finite-state model of PPM channel transmission.

4.1.3 I2C

I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line). Like SPI, I2C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master.
With I2C, data is transferred in messages. Messages are broken up into frames of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted.

Start Condition: The SDA line switches from a high voltage level to a low voltage level before the SCL line switches from high to low.

Stop Condition: The SDA line switches from a low voltage level to a high voltage level after the SCL line switches from low to high.

Figure 4.5: I2C Frame Structure

Figure 4.5 shows that the address frame includes a single bit at the end which informs the slave whether the master require to write data to it or receive data from the same.
If the master wants to send data to the slave, the read/write bit is a low

voltage level.
If the master requests data from the slave, the read/write bit is at high voltage level.

4.1.4 Complementary Filter

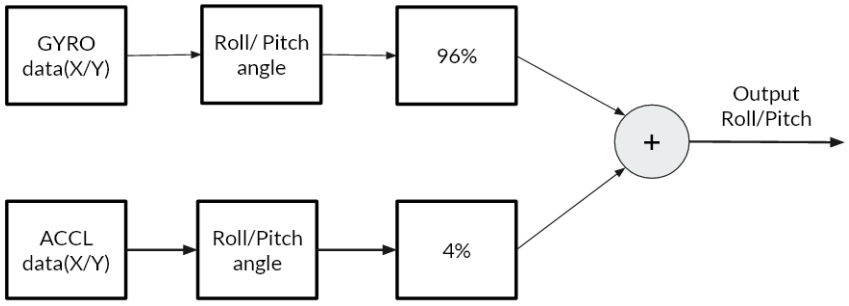


Figure 4.6: Complementary Filter - Logical Dataflow

Figure 4.6 shows logic followed for the implementation of the Filter design for gyrometer and accelerometer data from MPU-6050 [Complementary Filter]. The Gyro drifts off over the course of time not suitable for long term measurements. The accelerometer is more susceptible to vibrations so values over short time are not suitable. Here, a combination of both is obtained using this filter.

4.1.5 PID Loop

A proportional–integral–derivative controller is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control.
Here, PID is used to the stabilization of a quadcopter which is used to regulate its four basic movements: roll, pitch, yaw angles, and altitude. PID controllers offer a simple but effective solution to stabilize the aircraft because they make it possible to treat every variable independently within a limited range in which the behavior of the quadcopter is approximately

linear. Figure 4.7 shows that a PID controller continuously calculates an

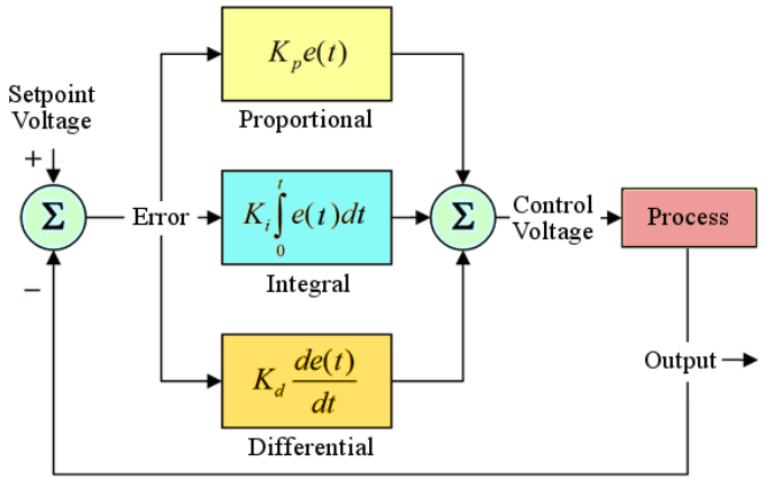


Figure 4.7: PID Controller loop

error value as the difference between a desired setpoint and a measured process variable and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name.

4.2 Feasibility Study

- For the FPGA Development board, the board chosen is Xilinx Virtex-6 based on few parameters like the scalability, optimal performance as per the extent of need for the project and cost compared to the other boards such as Xilinx Zynq 7000 or Altera Cyclone V available.
- MPU 6050 Inertial Measurement Unit is chosen as it is a 6 Axis IMU, has I2C communication and cost-effective compared to MPU 6500.
- IDE used for the simulations and the software realisation of the hardware design is Xilinx Vivado as the board chosen is Virtex-6 and moreover it is free upto an extent needed for this particular project.
- Verilog is chosen as the language of implementing the logic of the project as it is feasible and good for beginners.
- PID control loop is used for the control system as it is viable enough.

Sl. No	Requirement in the project	Hardware/Software available			Hardware/Software selection		
		Name	Features relevant to the project	Cost (₹) / License	Name	Reason	Cost (₹)
1	FPGA Development board	Xilinx Virtex - 6	Performance, Scalability	20000	Virtex - 6	.Cost .Optimal	20000
		Xilinx Zynq 7000	SoC, High Scalability	80000			
		Altera Cyclone V	SoC	25000			
2	IMU	MPU-6050	6-axis IMU, I2C	100	MPU - 6050	.Optimal	100
		MPU-6500	6-axis IMU, DMP, I2C	200			
3	IDE	Xilinx Vivado	For Xilinx boards	Free	Xilinx Vivado	.Virtex - 6	Free
		Quartus Prime	For Altera boards	Free			

Table 4.1: Feasibility Study 1

- The motor chosen is a BLDC one as it is both Efficient and Durable compared to a Coreless Motor available in the market.

Sl. No	Requirement in the project	Hardware/Software available			Hardware/Software selection		
		Name	Features relevant to the project	Cost (₹) / License	Name	Reason	Cost (₹)
4	Language	VHDL	Non-C Syntax, HDL	-	Verilog	.Feasible	-
		Verilog	C Syntax, Compact, HDL	-			
		SystemVerilog	C++ Syntax, HDL, HVL	-			
5	Control System	PID	Basic control system	-	PID	.Viability	-
		ML	Complex	-			
6	Motor (4)	Coreless	Low cost, compact	500	BLDC	.Efficiency .Durability	2000
		BLDC	Efficiency, Large torque	2000			

Table 4.2: Feasibility Study 2

Table 4.1 and Table 4.2 shows a summary of the options we had, choices we took, features of each of these for software and hardware.

Chapter 5

WORK SCHEDULE

The following is a brief on our plans to make progress.

Athul - Integrate PPM into PID

Medha - Integrate PWM into PID

Omar - Filter for MPU6050 and I2C protocol for MPU6050

Sebastian - Implement Kalman filter.

Then we will integrate everything into the PID together. Table 5.1 gives an account of the work done and the schedule ahead.

PROJECT TITLE	FPGA FLIGHT CONTROLLER						
PROJECT GUIDE	Mr. Jagadeesh Kumar P						
GROUP MEMBERS	Athul John Kurian (AJK),Medha Lakshman Rao(MLR),Omar Bin Shafi(OBS),Sebastian James (SJ)						
DATE	7th Jan 2021						
SL. NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	
1	Project Conception and Initiation						
1.1	Topic Selection	OBS, AJK,SJ, MLR	30th July			100%	
1.2	Literature Review	OBS, AJK,SJ	25th Aug			60%	
1.3	Feasibility Study	MLR,OBS,SJ	3rd Nov			70%	
1.4	Project Planning	OBS, AJK,SJ,MLR	12th Aug			90%	
1.5	Project Initiation	OBS, AJK, MLR	14th Nov			45%	
1.6	Scope & Goal Setting	SJ,AJK,MLR	1st Nov			85%	
1.7	Budget	MLR, AJK	30th Nov			40%	
1.8	Background study	SJ	7th Sept			65%	
2	Work Schedule						
2.1	Software installations	OBS, MLR, AJK, SJ	14th Sept			100%	
2.2	Getting to know the software	OBS, MLR, AJK, SJ	5th Oct			100%	
2.3	PPM Decoder	AJK	30th Nov			80%	
2.4	I2C Communication, filter	OBS, SJ	27th Oct			85%	
2.5	PID Loop	OBS, AJK	18th Sept				
2.6	PWM Encoder	MLR	22th Nov			90%	
2.7	Simulation	AJK, MLR, OBS		15th Mar			
2.8	Component Testing			10th Apr			
2.10	Debugging			20th Apr			
2.11	Finalising the project	SJ, AJK					
2.12	Novelty	SJ, AJK	30th Nov			90%	

Table 5.1: Work Done

Chapter 6

WORK DONE

- Athul
  - Familiarised with Vivado ISE; implemented gates, adders, clocks.
  - Learned basics of PID, learned PPM decoding.
  - Completed PPM decoding in Verilog, hardware implementation pending.

The following is the simulation output of a PPM decoder using Verilog:

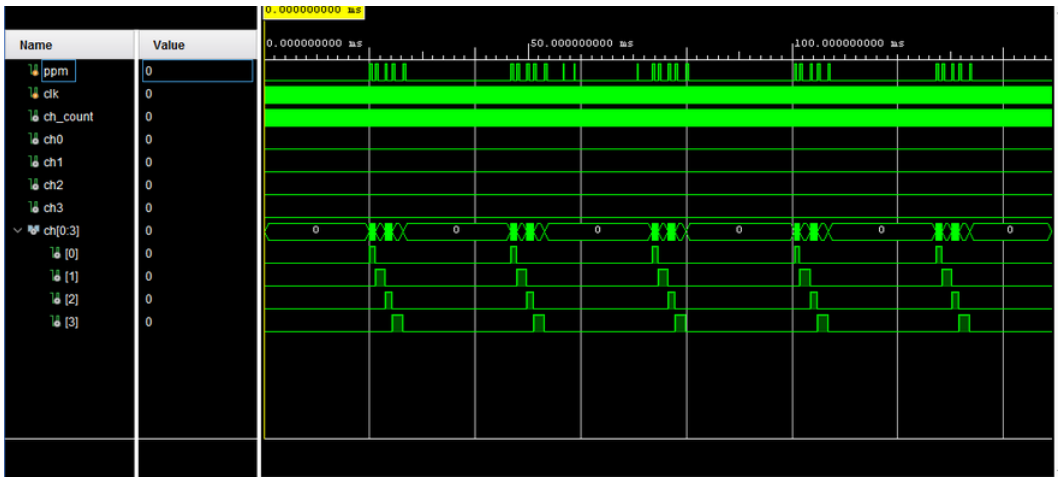


Figure 6.1: Pulse Position Modulation - Simulation Output

Here, 'clk' is the master clock of 1 MHz frequency, 'ppm' is the input PPM signal from the receiver, 'ch_count' is a pos-edge incremental counter which counts the clock signal, 'ch0', 'ch1', 'ch2' & 'ch3' are test variables used in Tcl console and 'ch' is a bus which consists of the decoded PPM

signal as 4 separate channels for flight movement control in the form of servo signals.

- Medha
  - Familiarised with Vivado ISE; implemented gates, adders.
  - Learned basics of PWM and implemented PWM encoding in Verilog.

  The following is the simulation output of a PWM encoder using Verilog:

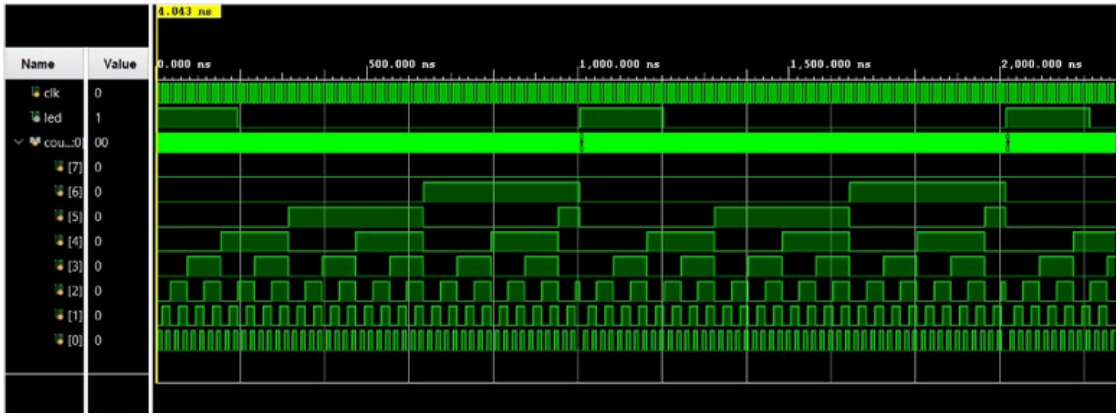


Figure 6.2: Pulse Width Modulation - Simulation Output

Here, 'clk' is the master clock, 'counter' is an incremental pos-edge counter which counts the clock signals that resets after a full PWM cycle is completed and 'led' is the PWM encoded output signal. The currently shown signal is a PWM signal with duty cycle of 20%.

- Omar
  - Familiarised with Vivado ISE; implemented gates, adders, clocks.
  - Learned filter design, started to implement filter.
  - Implemented I2C protocol.

  Here, 'clk' is the master clock, 'addr' is the decoded slave address bus, 'reg_addr' is the bus holding the decoded register address of the register inside the MPU-6050 sensor, 'data_in' is another bus which holds the decoded data transmitted into the FPGA from the sensor, 'enable' shows if the I2C signal line is currently in use or not, 'rw' shows if the mode set

is either read or write data, 'data_out' is a bus used to send out the data from the FPGA, 'i2c_sda'is the data line and 'i2c_scl' is the clock signal of the I2C protocol.

The following is the simulation output of I2C protocol using Verilog:

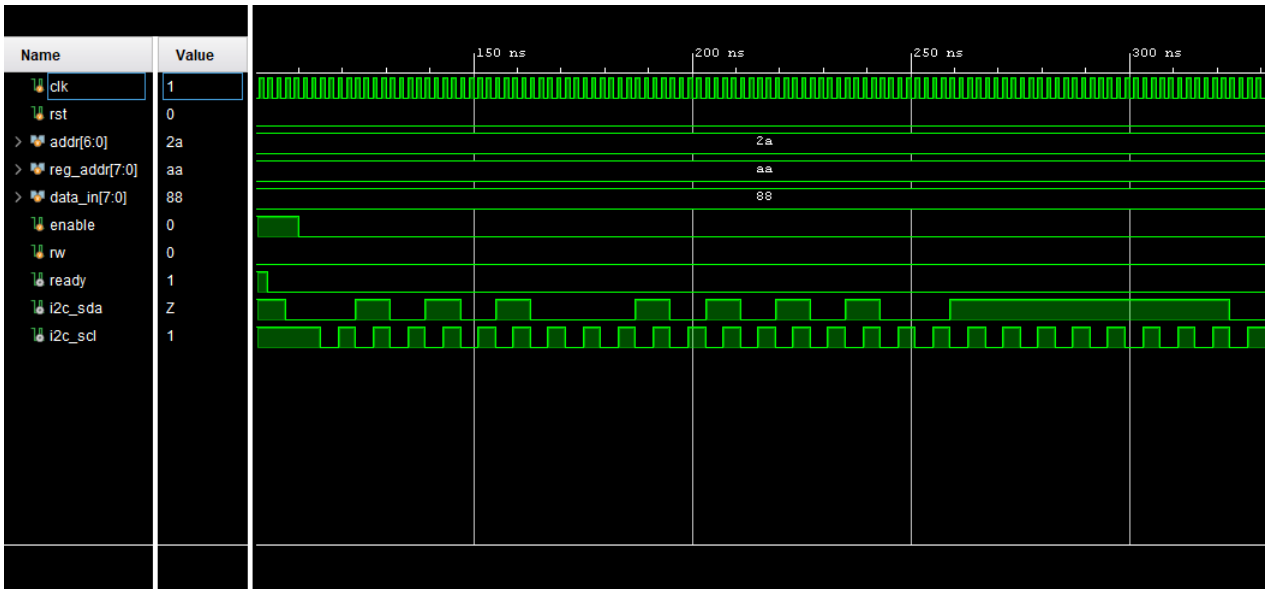


Figure 6.3: I2C - Simulation Output

- Sebastian
  - Learned the basics of FPGA development from Coursera.
  - Familiarized with Quartus Prime Lite.
  - Read the reference papers; 5 chapters from 3 texts.
  - Learned I2C implementation in detail.

Started the course Introduction to FPGA design for Embedded Systems offered by University of Colorado Boulder. Attended workshops to know more on FPGA design, referred various online resources to get clarity when in doubt regarding the flow of the project. Familiarized with Altera Quartus Prime Lite in an attempt to gain insight of how it works when compared to Vivado(which is for Xilinx boards) and save time in the future when it comes to implementing the project in hardware boards related to Altera. Referred standard journal, textbooks and conferences papers to share

the insight gained with teammates in order to incorporate ideas by getting to know various implementations, bring about novelty by knowing how much of variety already has been implemented, understand the basics of FPGA design, existing trends and follow a work path different from the rest of the teammates so as to be the one who could think in a different manner from the usual thought process, when in need.

Chapter 7

CONCLUSION

- The aim of this project is to Implement a Flight Controller on FPGA
- A PID controller is used to provide stability
- Integration of various sensors and Kalman filter into the flight controller to improve stability is one of the future scopes of the project.
- We also have plans to incorporate ML/FSM after completion of implementing PID controller in the system.
- Developing an MVP from the project is also one of the future scopes of this projet.

References

[1] N. Monterrosa, J. Montoya, F. Jarquín, and C. Bran, "Design, development and implementation of a uav flight controller based on a state machine approach using a fpga embedded system," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–8.

[2] B. L. Sharma, N. Khatri, and A. Sharma, "An analytical review on fpga based autonomous flight control system for small uavs," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 1369–1372.

[3] G. Premkumar, R. Jayalakshmi, and M. Akramuddin, "Design and implementation of fpga based quadcopter," in *2018 International Journal of Engineering Technology Science and Research, Premkumar2018DesignAI*, vol. 5, no. 3, 2018, pp. 1–5.