RISC V Based Drone Flight Controller
ECE 562 Project Proposal

Mitchell Dzurick - mitchdz@email.arizona.edu
Lena Voytek - dvoytek@email.arizona.edu
Amir Mohammad Asdagh Pour - asdaghpour@email.arizona.edu

2/17/2020

`https://github.com/mitchdz/ECE562`

Contents

1   Names and email addresses of group members

Mitchell Dzurick - mitchdz@email.arizona.edu
Lena Voytek - dvoytek@email.arizona.edu
Amir Mohammad Asdagh Pour - asdaghpour@email.arizona.edu

2   Topic Description

The goal of this project is to create a flight controller for a drone using a RISC V architecture. This includes purchasing and building a micro drone that will fashion a RISC V FPGA in order to create custom ISA subroutines for the drone. These custom subroutines will range from simple things such as move forward x feet to more complex things such as move to a certain GPS locaiton. The implementation of the more complex subroutines will get completed if time permits.

3   Motivation

The team taking on this project simply likes building and making drones. The project seems doable in a semester, but still offers a fun and challenging thing to solve. The goal here is not to make something super novel, but rather to have fun and learn a lot about RISC V ISA in the process.

4   Related work

4.1   Worlds smallest autonomous rotor drone

drone utilizing the GAP8 processor which is running 32-bit RISC-V to achieve a low power autonomous drone.

4.2   This tiny drone with a tiny brain is smart enough to fly itself

drone uses a mobile processor named GAP8. It packs eight processing cores optimized for running artificial intelligence applications.

4.3   Development of Drone Capable of Autonomous Flight Using GPS

An experimental drone was developed by equipping a microcomputer of Raspberry Pi 2.0 and a GPS sensor

5   Proposed methodology

The flight controller shall consist of an FPGA capable of booting verilog code. The codebase that is being looked at is ultraembedded's 32-bit RISC-V verilog implementation located at `https://github.com/ultraembedded/riscv` which can be compiled using Vivado. This allows us to create our own ISA and accompanying instructions to do very specific tasks. Before this is done, a drone needs to be constructed and flight tested with a known-working flight controller such as the Naze32 flight controller. Once a build is done, the FPGA will be loaded with the RISCV core and linux will be put on the core. An attempt to control the drone via the FPGA GPIO (General Purpose Input Ouput) ports will be done. After being able to control the drone simply through Linux, the core will be modified for special subroutines. To evaluate the project, the

6   Proposed Timeline

The following table shows a general outline for the project. These dates are of course subject to change, but are a guideline.

Date	Task	Member
2/21	Research Parts	Lena, Amir
2/21	Install software	Mitchell, Lena, Amir
2/21	Build RISC-V core	Mitchell, Lena
2/31	Purchase parts	Mitchell, Lena, Amir
3/6	Procure parts	Mitchell
3/13	Load Linux on FPGA	Lena, Mitchell
3/13	Base drone build	Mitchell, Amir
3/31	Get drone flying using Linux	Mitchell, Lena, Amir
4/10	Build custom ISA and instructions	Mitchell, Lena, Amir
4/24	Test all ISA implements	Mitchell, Lena, Amir

7   Anticipated Results

The results of this project is to have a working drone that has a small set of instructions and a custom ISA for certain subroutines.

8 References

Ultraembedded RISC-v
https://github.com/ultraembedded/riscv

Worlds smallest autonomous rotor drone
https://riscv.org/2018/05/fast-company-article-this-tiny-drone-with-a-tiny-brain-is-smart-enough-to-fly-itself

This tiny drone with a tiny brain is smart enough to fly itself
https://www.fastcompany.com/40575392/this-tiny-drone-with-a-tiny-brain-is-smart-enough-to-fly-itself

Development of Drone Capable of Autonomous Flight Using GPS
ttps://link.springer.com/article/10.1186/s40537-019-0214-3