

**ENSC 351**  
**Project**  
**Overview**

**October 11<sup>th</sup>, 2011**  
**Eric Matthews**

# Outline

Platform Overview

Project Outline

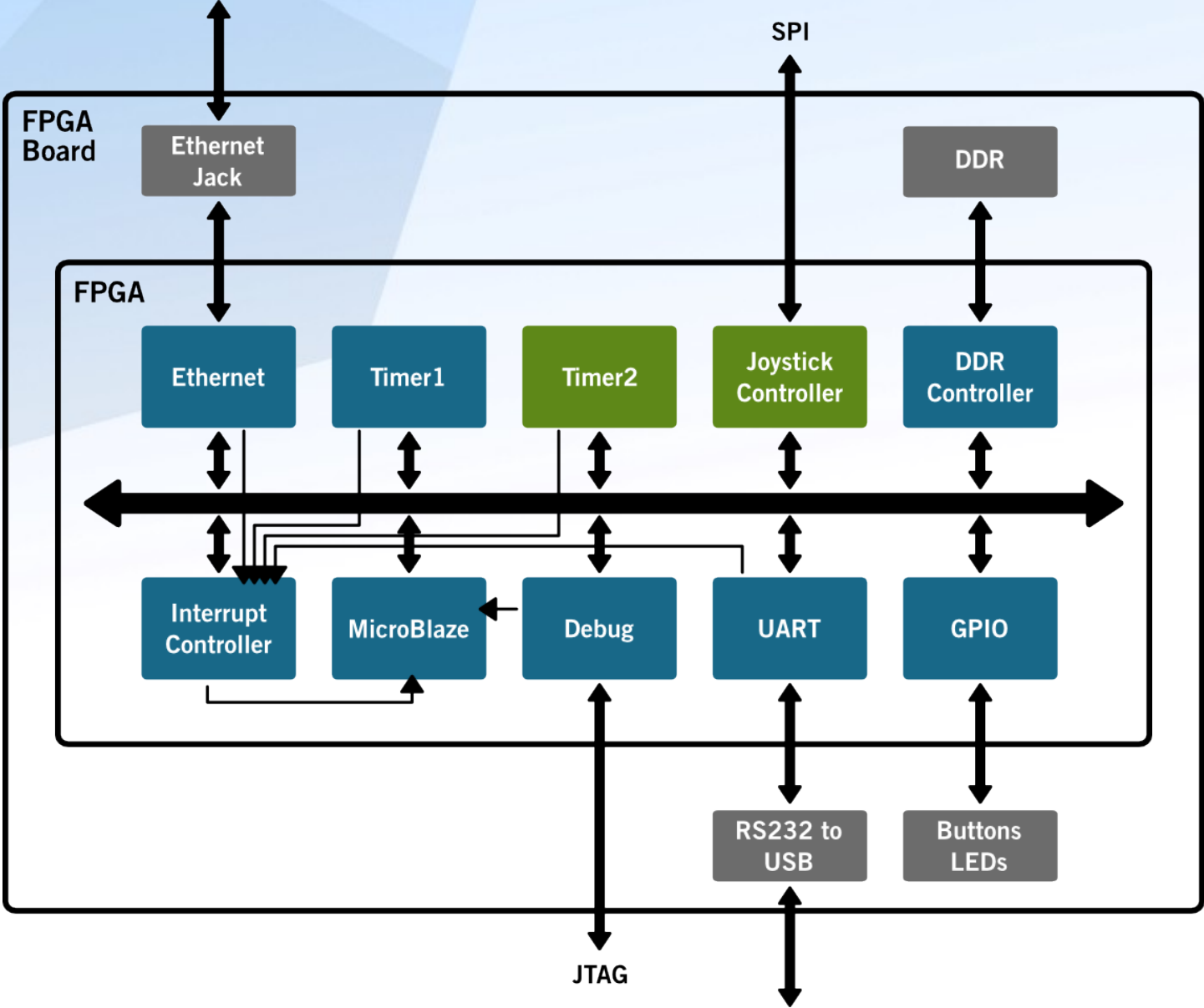
Project Timeline

Project Details

Demo

Questions?

# High-level Block Diagram



# MicroBlaze

Simple Reduced Instruction Set Computing (RISC)  
based architecture

Harvard architecture

Direct mapped L1 caches 32kB

Most instructions single cycle

# Running PetaLinux

1. Program FPGA through JTAG
2. FS-BOOT starts running
3. Download Linux system image into DDR
4. Execution passed to Linux kernel

# Linux System Image

Contains complete filesystem

DDR partitioned into “diskspace” and memory

Full Linux kernel (2.6.37)

Current kernel (3.0 --> 2.6.40)

MicroBlaze support in mainline kernel

System has a trimmed down set of userspace libraries

Similar to what would run on an ebook reader

# Running PetaLinux

1. Program FPGA through JTAG
2. FS-BOOT starts running
3. Download U-BOOT bootloader into DDR
4. U-BOOT fetches kernel image through ethernet and places image in DDR
4. Execution passed to Linux kernel

# ~~System Limitations~~

## Project Challenges

66MHz Processor Frequency

Limited Physical memory

RISC based processor

Limited caches



# Design Considerations

Kernel <--> Userspace context switching

Cache friendly algorithms

Scalability

1. ( )  
2. ( )  
3. ( )  
4. ( )

# Snake Game

Create a console based “snake” game

Work within constrained resources

Learn to write Linux device drivers



# Snake Game

Takes place on a torus grid-based playing field

Level configuration files loaded at startup

Includes obstacles and snake starting points

# Division of Work

Work divided into two streams

Each stream will be assigned to one pair

Once chosen, pairs cannot swap streams

# Tasks

<b>Group 1 (Pair A) Work</b>	<b>Group 2 (Pair B) Work</b>
Stream 1 Task 1 (demo)	Stream 2 Task 1 (demo)
Integrate	
Stream 1 Task 2 (demo)	Stream 2 Task 2 (demo)
Integrate	
Stream 1 Task 3	Stream 2 Task 3
Integrate (FINAL DEMO)	

# Tasks

## Stream 1

Joystick Driver & Basic Game

Loading configuration files, GUI and 2 player mode

Advanced AI: Highest Score per Unit Time

## Stream 2

Timer Driver

Basic AI (a collection of different algorithms)

Advanced AI: Longest Length



# Marking

Given 2 weeks per task

Marked as a pair

Marks assigned for completion and ability to answer questions

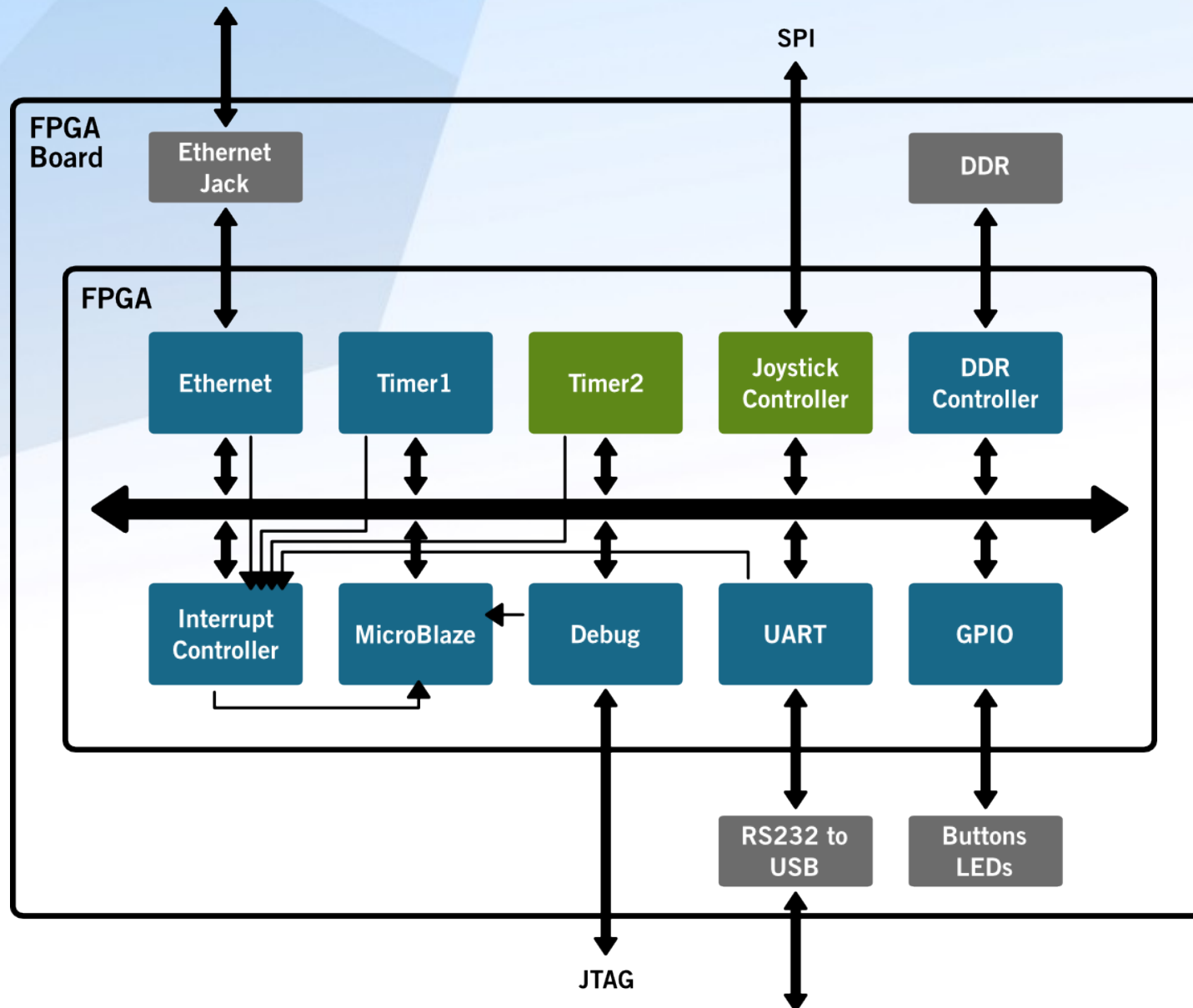
# Task 1

Gain experience writing device drivers for the Linux kernel

Write custom drivers for joystick and timer peripherals

Prepare basic framework for snake game

# High-level Block Diagram



# Task 1 (Basic Game)

The foundation for the

Create “empty” playing field

Move snake through keyboard input

Design structures and algorithm for snake movement

# Task 2

## Stream 1

Finish-up framework for game

Load configuration files

## Stream 2

Creating basic AI

(Greedy algorithm, avoid obstacles, switchback)



# Task 3

## Stream 1

Highest Score AI per Unit Time

Collect food as quickly as possible while surviving for the longest time

## Stream 2

Longest Length (Surviving for the longest time)

# Demo



# Questions?