CS5780

Administrative Details

Very lab and project heavy

Labs are designed to be done in the specified times.

Lectures are to prepare for labs.

Not a programming class – TAs will help out with that stuff.

Don’t be afraid to bug the instructor or TAs!!!!

Communicate, communicate, communicate!

[Pierre-emmanuel.gaillardon@utah.edu](mailto:Pierre-emmanuel.gaillardon@utah.edu)

Office hours: after hours and by appointment

Textbooks (optional): The Definitive Guid to ARM Cortex-M0 and Cortex-M0+ Processors, 3rd Edition, Joseph Yiu; Introduction to Embedded Systems: A Cyber-Physical Systems Approach, Edward Lee/Sanjit Seshia (downloadable at leeseshia.org)

2 parts in the class:

Lectures, lab sessions, and homework

Graduate presentations and mini-projects

Mostly for graduate students.

You **must** partner with some in your lab section.

Best if it is cross discipline pairs

Also best if it’s the same partner throughout the semester

Labs begin as marked on Canvas

Wednesday through next week will be for help sessions on homework.

Labs are designed to be able to do from home.

There is hardware to buy for this class; check slides for them.

MUST BUY

Make sure to use antistatic mats and connectors in lab when handling the boards

Or ground yourself

All due dates are on Canvas

Lab reports are due on Canvas by the time of your next lab section

Make sure to have the lab section number on all reports

Strict time submission rule on Canvas – more so for homework than labs

Late submissions are at risk for not being graded.

Students pass the class so long as they: don’t cheat, turn homework in on time, and generally *do what they’re supposed to*

In-Person classes will not be recorded (as of 2022-01-11)

Everything else can be done asynchronously.

L1 – Embedded Systems

What are Embedded Systems?

Computers whose job is not primarily information processing, but rather is interacting with physical processes.

Grinding numbers and doing something with it.

Embedded systems vs. general computing

Time matters

As fast as possible is not always good enough

Concurrency is intrinsic

It’s not an illusion (like time sharing)

It’s not necessary about exploiting parallelism.

Processor requirements can be specialized

Predictable, repeatable timing

Support for common operations

Need for specialized data types

Programs need to run (essentially) forever:

Memory usage must be bounded

Rebooting is not acceptable

L. Memory

A processor only sees memory!

What does a processor do?

It manipulates data!

Where are the data located?

In memory!

How does it access memory?

Through a data bus, with access and data.

2-8-2022

Watch the lecture videos!!!

2-24-2022

* Serial – protocol family
* Parallel – protocol family
* WiFi
* RF (phone)
* Protocols
* SPE
* Bandwidth
* Pattern bits
* Clocks

How do systems communicate?

There are many different considerations and solutions.

It depends on the communication medium used.

03-29-2022

Exercise

Water flow through pipe should maintain a steady velocity above or around a standard value for seamless usage at homes and other commercial places. With long complex structures of piping systems used in commercial usage, there are chances that defects start to arise resulting in an unusable water flow.

In these exercises, we are going to design a monitory system for water flow through pipe. When this system detects a problem, it will assert a warning signal to the maintenance team to check the pipe. Signal from a Water Flow Probe will be used by the system to monitor the water velocity in pipe against a minimum and maximum threshold value.

Assume the following:

There is an ADC that converts the velocity value from the water flow probe to an 8-bit number, **data**

This variable is an output of the ADC

The ADC is started by a high pulse on the **adcStart** signal

It is a pure signal.

It is enabled by writing a 1 into the ADC start register.

Essentially the triggering event.

The ADC indicates completion by a rising edge on the **adcDone** signal.

An issue in the pipe is assumed when **100 consecutive violations are detected.**

Once the issue is confirmed, a **warning** is indicated by setting an I/O port pin to high.

**Preamble question:** What is the type of each signal?

**State Machine Modeling:** Draw an extended state machine diagram for the monitoring system. You may add any other variables that you need. For simplicity, you can assume that the ADC delay and software delays are negligible compared with the waiting time between readings.

Start with the variables and inputs given.

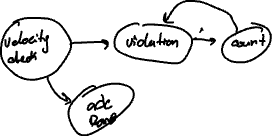
Relationship between the states.

Either 2 or 5 states

Water flows in and the monitor will check to see if it is within the threshold value.

If it is not within the threshold value, start counting the violations.

The violations are determined by…?



**Additional questions:**

When does a reaction occur in your model?

What is the associated model type?

How to reinitialize the system?

Both adcDone and adcStart are pure signals.

adcStart is an input signal.

How to start a conversion in your ADC?

Enable the start bit by writing a 1 into the adcStart bit register

This is essentially the triggering event.

adcDone is in an output signal.

It could be the status register (but it is purely passive).

You can trigger an interrupt to emulate an adcDone signal.

Easier to do.

While loop to read the status register

Loop through until the register changes

Asserting means it is sending a signal (?)

Mainly a pure signal (?)

Because the assertion only happens after 100 consecutive violations are detected.

A variable to hold that value is required.

Probably a statement to

On the exam, write your assumptions!!!

L20. Transactions

Race conditions

The database does not become corrupt

One transaction will succeed while the other will violate PK constraint

DBMS automatically protects against race conditions

Transactions

Only have 2 possible outcomes

0% completion

100% completion

Guaranteed even with concurrent transactions

ACID

Atomicity

Operations either completely succeed or completely fail

Consistency

Operations never result in violated constraint

Isolation

Durability