Welcome to ECE251! Introduction to Microprocessors

• Tuesday, August 21
• What this course is about: Syllabus, Labs, etc.
• Review of Number Systems, including simple math
  – You learned this in ECE102 or its equivalent.
• Introduction to processor instructions
• Read: Chapters 1 and 2 in Text

• Labs: YES, THEY STARTED YESTERDAY!
  – #1 is an important and easy introduction of lab tools. Due next week!
  – See web page under “Lab Assignments” (and look at Lab #2 now!)
• Homework: #1 Due Sept. 6. See web page under “Homework”.
• Web Page: (Bookmark this!)
  www.engr.colostate.edu/ECE251/course_info.html
  – This lecture (and all lectures) are on our web page ahead of time.
Just for Fun

1. Figure out the rules (inductive)
2. Solve it (deductive)
EE 251 – Introduction to Microprocessors

Course Syllabus: Fall 2018

Instructor: Dr. Bill Eads, Engr C103E
Office phone: 491-0717 (during office hours only)
Home phone: 667-6914 Phone calls welcome (until 10 pm!)
email: EadsinCO@gmail.com
Office hours: Tues 9:30-10:30; Thurs 9:30-11 or by appointment

Lab TAs: Joel Kraft
          Emily Dalton
          jkraft@rams.colostate.edu
          emily.dalton@rams.colostate.edu

HW Grader: Jim Bryce
           jbryce@southeastoffset.com

Text: *Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C*, Yifeng Zhu
      ISBN 978-0-9826926-6-0
Course Description: Microprocessor organization, assembly language, I/O techniques, real-time interfaces, applications, hardware and software.

Prerequisite: ECE102 (Digital Circuit Logic)

Grading and Exams: Midterm Exam 20% 
PRELIMINARY Final Exam 25%  
((+/ grading used) Lab & Practicals 20%  
Grading is curved Homework Assignments 10% 
Quizzes & Participation 25%  
}

Grading is curved 45%

Homework Schedule: Homework problems will usually be assigned every week or two and will be turned in to the white ECE251 box in the BC Infill. Late homework will not be accepted without prior instructor approval.

Labs: There will be a series of 9 labs, typically due each week. There are two lab practical exams in lab during the course, focused on programming skills. Successful completion of all labs is required for a passing course grade.

You are expected to work on all homework problems and labs yourself (or within your team for a few labs), but reasonable collaboration is allowed and encouraged.
Attendance (Physical and Mental)
(Yes, I know how early 8 o’clock is)

**Attendance in class:** EXPECTED. No makeup on missed quizzes is allowed, but your lowest grade will be dropped. Alertness in class matters significantly—do what works for you to be alert at 8 a.m. If you do not attend class regularly and alertly, you will probably **not** pass this course. Since this is a required course for many of you, that would mean you have to take it again. Don’t do that to yourself!

**Quizzes:** Quizzes will be given every week or two. Some will be scheduled; some will not be announced. You may **not** collaborate on quizzes. Note that quizzes are a **quarter** of your grade: more than the mid-term exam, more than labs, and more than homework. So SHOW UP and PAY ATTENTION!
ECE 251 Course Outline

Number Systems & Digital Logic Fundamentals

Microprocessors: Major Components

ARM Cortex-M4 Microcontroller: Register Model & Memory Addressing

ARM Cortex-M4 Assembly Language Programming

ARM Cortex-M4 Instruction Set:
  Data transfer and manipulation instructions
  Arithmetic Instructions
  Logical and Bit Operations
  Branch Instructions

Assembly Process & Advanced Assembly Programming
  Software Delay
  Programming Techniques
  Loops
  Stack and Stack Pointer
  Subroutines and Parameter Passing
 Exceptions—Reset and Interrupts

General Purpose (Parallel) I/O

MID-TERM Exam about here (Probably Thurs. Oct. 4)

ARM Cortex-M4 Serial Communication Methods

Using Interrupts – SysTick as a Real Time Clock

Standard Timer Module

Fixed and Floating Point Number Representations

ARM Cortex-M4 Analog-to-Digital Converter System

Adding Memory to a Processor

Final Exam: Tuesday, December 10, 2:00 pm, 1 to 1¼ hours
# ECE 251 Timing 2018

<table>
<thead>
<tr>
<th>Class #</th>
<th>Lecture Topic (Chapter)</th>
<th>Lecture/Due Date</th>
<th>Week</th>
<th>Lab #</th>
<th>Lab Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro, Numbers (2)</td>
<td>8/21</td>
<td>Week 1</td>
<td>1</td>
<td>SW Setup</td>
</tr>
<tr>
<td>2</td>
<td>ARM Microcontroller Components, etc.</td>
<td>8/23</td>
<td>Aug 20</td>
<td></td>
<td>Assy Prog’g</td>
</tr>
<tr>
<td>3</td>
<td>Loading and Storing Data (5)</td>
<td>8/28</td>
<td>Week 2</td>
<td>2</td>
<td>Adv Assy</td>
</tr>
<tr>
<td>4</td>
<td>Arithmetic and Logic (4)</td>
<td>8/30</td>
<td>Aug 27</td>
<td></td>
<td>Branch</td>
</tr>
<tr>
<td>5</td>
<td>Assembly Language and Assembler (3)</td>
<td>9/4</td>
<td>Week 3</td>
<td>3</td>
<td>Simple Progs</td>
</tr>
<tr>
<td>6</td>
<td>Branching and Looping (6)</td>
<td>9/6</td>
<td>Sept 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Stack and Subroutines (8)</td>
<td>9/11</td>
<td>Week 4</td>
<td>4</td>
<td>Subroutines and Stack</td>
</tr>
<tr>
<td>8</td>
<td>Passing params to sub via reg (8.5)</td>
<td>9/13</td>
<td>Sept 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Recursive sub calls (short) (8.7)</td>
<td>9/18</td>
<td>Week 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Parallel I/O (14)</td>
<td>9/20</td>
<td>Sept 17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Finish Parallel I/O; Interrupts Intro</td>
<td>9/25</td>
<td>Week 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Interrupts and Resets (11)</td>
<td>9/27</td>
<td>Sept 24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Midterm review</td>
<td>10/2</td>
<td>Week 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><strong>Midterm exam</strong></td>
<td>10/4</td>
<td>Oct 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Serial I/O (22)</td>
<td>10/9</td>
<td>Week 8</td>
<td>5</td>
<td>Practical #1</td>
</tr>
<tr>
<td>16</td>
<td>SPI, LCD, Logic Analyzer, Lab 5</td>
<td>10/11</td>
<td>Oct 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>ATD (20)</td>
<td>10/16</td>
<td>Week 9</td>
<td>6</td>
<td>SPI</td>
</tr>
<tr>
<td>18</td>
<td>ATD continued + 7-segment display</td>
<td>10/18</td>
<td>Oct 15</td>
<td></td>
<td>ATD</td>
</tr>
<tr>
<td>19</td>
<td>SysTick Real Time Clock (12)</td>
<td>10/23</td>
<td>Week 10</td>
<td>7</td>
<td>Systick Timer</td>
</tr>
<tr>
<td>20</td>
<td>Timer Module (15)</td>
<td>10/25</td>
<td>Oct 22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Timer Module continued</td>
<td>10/30</td>
<td>Week 11</td>
<td>8</td>
<td>Timer &amp; Systick Clocks</td>
</tr>
<tr>
<td>22</td>
<td>Fixed- and Floating-point Arithmetic (11)</td>
<td>11/1</td>
<td>Oct 29</td>
<td></td>
<td>Period, Duty</td>
</tr>
<tr>
<td>23</td>
<td>Instruction Encoding &amp; Decoding (13)</td>
<td>11/6</td>
<td>Nov 5</td>
<td>9</td>
<td>Cycle of sigs</td>
</tr>
<tr>
<td>24</td>
<td>UART (22.1)</td>
<td>11/8</td>
<td>Week 12</td>
<td></td>
<td>Individual</td>
</tr>
<tr>
<td>25</td>
<td>UART continued; USB Overview (22.4)</td>
<td>11/13</td>
<td>Nov 12</td>
<td></td>
<td>Projects</td>
</tr>
<tr>
<td>26</td>
<td>Memory I</td>
<td>11/15</td>
<td>Week 13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Memory II</td>
<td>11/27</td>
<td>Nov 26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Memory III</td>
<td>11/29</td>
<td>Week 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Guest Speaker: Future of Computers</td>
<td>12/4</td>
<td>Dec 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Final Exam Review</td>
<td>12/6</td>
<td>Week 15</td>
<td></td>
<td>Practical #2</td>
</tr>
<tr>
<td></td>
<td><strong>FINAL EXAM</strong></td>
<td>12/10</td>
<td>2:00 pm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ECE 251 Lab

We will use the Texas Instruments Tiva C Series TM4C123G LaunchPad Evaluation Kit.
ECE 251 Lab (continued)

Each student will be provided a LaunchPad board.
• Board replacements will be available for broken boards, but you are responsible for the cost of any required replacement (about $12).
• Boards will be available through your Lab TA starting this week.

You will work as INDIVIDUALS on most, but not all, lab projects
✓ Writing Programs
✓ Wiring Hardware
✓ Demonstrating requested functionality

You will have a total of 9 lab assignments.
Lab 1 started this week and is due next week at beginning of your lab time.
—This lab is important. It will introduce:
  • Our processor instruction set,
  • Our development board,
  • Our software development environment.

You must successfully complete ALL lab projects to pass the course!
## ECE251 Lab Schedules 2018

<table>
<thead>
<tr>
<th>Lab #</th>
<th>Day</th>
<th>Time</th>
<th>TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>L07</td>
<td>M</td>
<td>9-11:50</td>
<td>Emily</td>
</tr>
<tr>
<td>L08</td>
<td>M</td>
<td>4:30-7:20</td>
<td>Joel</td>
</tr>
<tr>
<td><strong>L09</strong></td>
<td>T</td>
<td><strong>10-12:50</strong></td>
<td><strong>Joel &amp; Jim</strong></td>
</tr>
<tr>
<td>L02</td>
<td>T</td>
<td>2-4:50</td>
<td>Joel</td>
</tr>
<tr>
<td>L01</td>
<td>W</td>
<td>12-2:50</td>
<td>Emily</td>
</tr>
<tr>
<td>L05</td>
<td>W</td>
<td>5-7:50</td>
<td>Joel</td>
</tr>
</tbody>
</table>
ECE251 Office Hour Options

TBD for TAs and Grader
Some Helpful Tools
But Not Required

You might find it beneficial to have a 3+ digit voltmeter, wire strippers, diagonal wire cutters, needle nose pliers, and (maybe) a soldering iron.
Enough!
Let’s Get Going!
Brief **Review**--Number Systems (see text: 2.2-2.4)

We can view a number as represented by:

\[
d_2 \ d_1 \ d_0 \ . \ d_{-1} \ d_{-2} \ d_{-3} = d_2 a^2 + d_1 a^1 + d_0 a^0 + d_{-1} a^{-1} + d_{-2} a^{-2} + d_{-3} a^{-3}
\]

where ‘a’ is the number base we use for this representation and \(d_i\) is a digit in this number base: \(0 \leq d_i \leq a-1\)

For example, with BINARY:

\[
b_2 \ b_1 \ b_0 \ . \ b_{-1} \ b_{-2} = b_2 2^2 + b_1 2^1 + b_0 2^0 + b_{-1} 2^{-1} + b_{-2} 2^{-2}\]

and \(b_i\) is 0 or 1

This is an excellent representation for digital systems, but poor for us to use. Why?
Hint: Quick, what is 100101111010\(_2\)? Bigger than 100? 1,000? 1,000,000?

Better choice for people:

**OCTAL** (2\(^3\)) or **HEXADECIMAL** (or **HEX**) (2\(^4\))

Just a grouping of binary bits into groups of 3 or 4 bits.
Straightforward for people to deal with. Why?
One-to-one representation of what’s happening inside the circuit.
Number Systems (cont’d)

E.g.: \[10010111010_2 = 100 \ 101 \ 111 \ 010_2 = \underline{___________} \ 8\]

\[= 1001 \ 0111 \ 1010_2 = 0x\underline{___________} \ (0x \text{ means \ HEX})\]

\[251 = 11111011_2 = 1111 \ 1011_2 = 0xFB\]

HEX advantage: Common bit lengths in computers (e.g. 16, 32, 64, 128 bits) are exact multiples of 4. I.e. they can be represented by complete hex digits.

E.g. A 16-bit computer has a 16-bit address, which is represented by exactly 4 hex digits (4x4 = 16).

How many octal digits would it take to represent a 16-bit address? \[\underline{_______}\]

A 32-bit address can be represented by exactly \[\underline{_______}\] hex digits.

In ECE251 we use HEX as our representation of choice since our processor, ARM, uses it and because just about the whole computing world now uses it.
2’s Complement Representation

• Let’s use an 8-bit example: $b_7\ldots b_0$
  – number = $-2^7\cdot b_7 + 2^6\cdot b_6 + \ldots + 2^0\cdot b_0$
  – Therefore, if high-order bit is 0, number is a positive seven-bit number: 0100 0101 = 0x45 = 69.
  – Also, if high-order-bit is 1, the number is negative, but with an offset of $-2^7 = -128$.
    • E.g. 1100 0101 = -128 + 0x45 = -128 + 69 = -59.
    • A quick way to compute the magnitude of a negative number is to take the negative of that number. Algorithm:
      – Complement each bit
      – Increment (add one to) the result.
      – E.g. 1100 0101 → 0011 1010 + 1 = 0011 1011 = 0x3B = 59
      – Sure enough, the negative of -59 is 59!

• Done! I said it would be a brief review!
ARM Cortex-M4 Instruction Set

• These are the instructions our processor uses to perform tasks

• Called **Thumb-2**. Includes
  – 16-bit instructions for small program size
  – 32-bit instructions for high performance

• Several classes of instructions. E.g. memory reference instructions and math instructions

• Will go through the key Thumb-2 instructions over the next few weeks.

• Meanwhile, to give you a feel for these instructions, a simple example follows:
# ARM Cortex-M4 Instruction Example

<table>
<thead>
<tr>
<th>LABEL</th>
<th>OPERATION</th>
<th>OPERAND</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>mov</td>
<td>r2,#11</td>
<td>; move 11 into register #2</td>
<td></td>
</tr>
<tr>
<td>add</td>
<td>r4,r2,#0xC</td>
<td>; [r2] + 0xC \rightarrow (goes into) r4</td>
<td></td>
</tr>
</tbody>
</table>

- **mov** means move
- r2 is register number 2
- #11 means the number $11_{10}$ (base 10)
- **add** means add
- r4 is register number 4
- #0xC is the hex number 0xC or $12_{10}$
- Comments follow “;” and explain what is happening
- [r2] means contents of register number 2
- What is the content of r4 (Hex) after these instructions?
KEIL μVision MDK

• KEIL is the brand name that ARM uses for its software for developing ARM code
• μVision (often written uVision) is the product name for this software
• MDK stands for Microcontroller Development Kit
  – Project Manager (keeps all software files organized)
  – Assembler (for our Thumb-2 instructions) and compiler (for C language which we WILL NOT use)
  – Editor for creating and editing assembler code (text)
  – Debugger
    • Run program, single step program, run to a breakpoint,…
    • Examine and change register and memory values…
Lab 1 Starts This Week (YESTERDAY)

When you complete this lab, you should be able to:

- Understand the fundamentals of assembly programming
- Recognize and use a few basic Thumb-2 instructions
- Understand Keil μVision (a tool for embedded software developers who write software in assembly language or C for microcontrollers).
- Connect to our development board (TM4C123G) and download a program using Keil μVision.
- Run a program and examine changes to memory and registers.
- Use breakpoints and stepping to debug a program.

Because you will perform all of these procedures in every lab, a complete understanding of the material in this lab is necessary.
Aren’t We Starting Off Pretty Fast?

Yep! We need to, because there’s lots to do, and this μVision – TM4C123G environment will be used from now on. It’s best to get used to it as quickly as possible starting with simple tasks.

This week is “special” for lab attendance. If you can’t/didn’t come to your assigned lab, come some other day. E.g. If you can come today but aren’t in the Tuesday lab, you’re welcome to come. AND if you have a Windows laptop, please bring it to lab this week.

Work with your lab TA to be sure you can attend SOME lab each week, especially if you have personal or class schedule conflicts in a specific week. But attend your assigned lab whenever possible.
Lab 9 Starts Now, Too

This lab is new this year and is your own independent project

- Already on our Lab web page
- Several suggestions on what you could choose
- Many different input and output devices are suggested, including some data sheets for those devices
- Grading on this lab is a function of its difficulty; work with your TA

I recommend you start thinking about what you’d like to do soon

- Discuss with and get approval from your TA before you begin (and I’m happy to discuss with you as well), and write your proposal.
- Perhaps order parts.
- Only one week of lab at end of semester focused on this lab; need major progress before then.
Questions?

Next Lecture will be on

- Microcontrollers and their components
- Our Processor’s Register Model
- How Programs are Executed
- Short Review of State Machines (ECE102)
- Maybe more on number representation-if time permits

Read Chapters 1, 2, 3.1, 3.2, and 3.4 in text

Research the following regarding ARM:

- What do these letters stand for?
- What do the letters in the “R” word/acronym stand for?
- ARM is a business. Who owns ARM? What do they make?
- These questions on a quiz are fair game!