Software Engineering 2DA4

Slides 1: Introduction

Dr. Ryan Leduc

Department of Computing and Software
McMaster University

Material based on S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design, 2nd Ed.*

How to Improve your 2DA4 Mark

- Show up for ALL classes and take good notes.
- Do all assignments and labs yourself.
- Follow up on what you get wrong/don't understand.

How to Improve your 2DA4 Mark: Long Version

- You can't learn something until you realize you don't already know it.
- Show up for **ALL** classes.
  - 10-15% of material given only in class. Missing class will drop mark by that amount, if you don't simply fail.
  - Material given in simpler form in class + examples explained + instructors experience.
  - If you don't understand part of material, you can ask questions.

How to Improve your 2DA4 Mark: II

- Pay attention in class and take good notes.
  - If you talk etc in class, you are missing the material: (10-15)% off your final mark minumum.
  - Print out course notes in advance. Copy down examples from board.
  - Write down anything that sounds important Prof says that is not in notes. You won't remember the details a month later when you go to study!

How to Improve your 2DA4 Mark: III

- Do assignments and labs yourself.
  - The assignments and the labs are where you learn the details of the material.
  - They give you practice and feedback on which parts you understand.
  - Exams build upon the assignments and labs. If you didn't do them, you won't be able to do similar questions in the exams.
  - Copying decreases your mark. Copying assignments might raise your final mark by 1-2%. However, you won't be able to do the questions on the exams which will cost you about 15-20%!

How to Improve your 2DA4 Mark: IV

- Ask questions!
  - If you don't understand something, don't ignore it. It will be on the test!
  - Ask your classmates about it, and if that fails, the Prof.
  - Always check solutions for assignments and exams. If you get it wrong, learn why, and how to do it right. You will see it again!

How to Improve your 2DA4 Mark: V

- Manage your time.
  - Evaluate the work to be done, and create a schedule.
  - Allocate your time. Don't leave things till the last minute.
  - Plan your studying. Start a few hours a day, weeks in advance. Plan what to study, on what day, and for how long.
  - For exam, scan all questions first. Do the easiest first.
  - Don't get bogged down. If you get stuck on a question, leave it and come back.
  - Read all parts of a question CAREFULLY before starting. Pay attention to details.

Course Motivation

Why should a software engineer study *digital logic design?*

- Digital logic provides the basis for a wide range of electronic devices, from calculators, watches, to televisions. Something so ubiquitous is worth knowing about.
- Digital logic provides the basic building blocks for all computers and embedded devices that software runs on.
- Understanding how the hardware works can help you write more efficient software.
- Embedded programs as well as OS device drivers have to interact with low level devices. At this level, software can damage hardware!

Course Motivation - II

- Assembly language programming will be more comprehensible.
- Compilers need to understand about hardware to create optimized code.
- Software useless without hardware. Someday you may need to design custom hardware to run your software!
- If developing software for new custom designed hardware, you need to be able to tell if the bugs are in your code or the hardware.

Digital Logic Introduction

- In a logic circuit, signals are restricted to a few discrete values.
- We are concerned about binary (can take on value of 0 or 1) logic circuits.
- Each signal in circuit can be represented by a binary digit, and thus such circuits are called "digital" circuits.
- In contrast, analog circuits take on a continuous range between a minimum and maximum value.

Purpose of Digital Circuit

- Take binary input signals and perform logic operations, possibly using state information (knowledge about past values of the circuit), to create binary output signals.
- Example: A circuit to add two numbers together.
- The two numbers are entered as binary numbers so that the circuit can understand them.
- Logic operations are performed (chapter 5 has details) on the two numbers to add them together, and the results are given as binary numbers.

Purpose of Digital Circuit

- Take binary input signals and perform logic operations, possibly using state information (knowledge about past values of the circuit), to create binary output signals.
- Example: A circuit to add two numbers together.
- The two numbers are entered as binary numbers so that the circuit can understand them.
- Logic operations are performed (chapter 5 has details) on the two numbers to add them together, and the results are given as binary numbers.

Purpose of Digital Circuit - II

- Often a digital logic circuit takes information in a form humans (external world) can understand, and converts it to binary format that it can use.
- Circuit then performs logic operations to do something "useful" (ie. addition).
- It then converts the results back into a form that humans (external world) can understand.

Interfacing with Digital Circuits

- Devices like keyboards can be used to allow numbers to be entered as decimal numbers, then an additional logic circuit is required to convert the numbers to binary.
- Display hardware like a seven segment hex display can be used to display the results of the adder circuit.
- Circuit to convert from binary to control signals for the display are required.

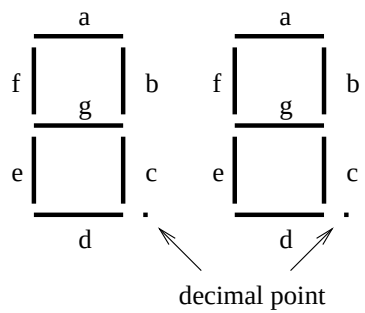
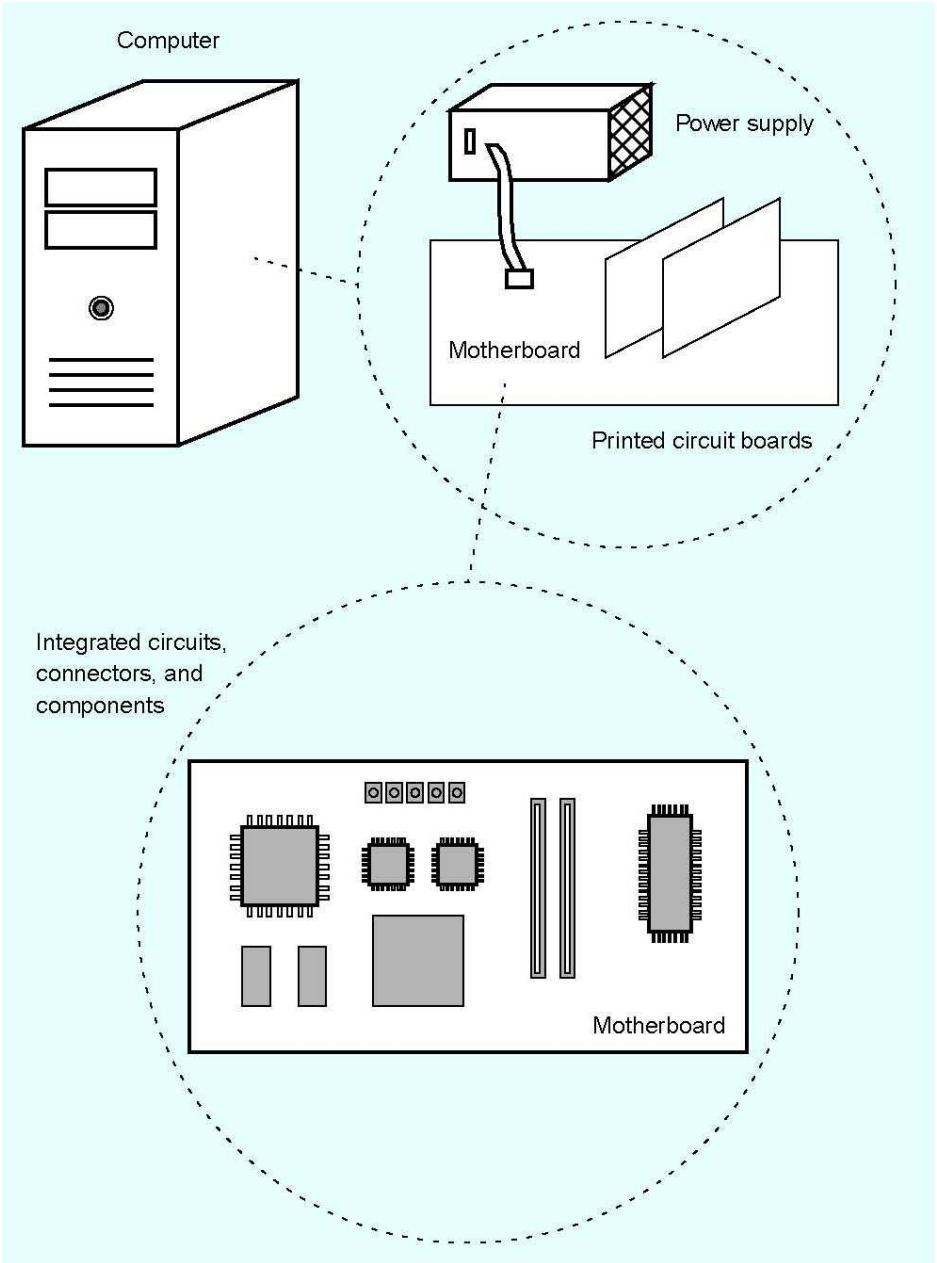
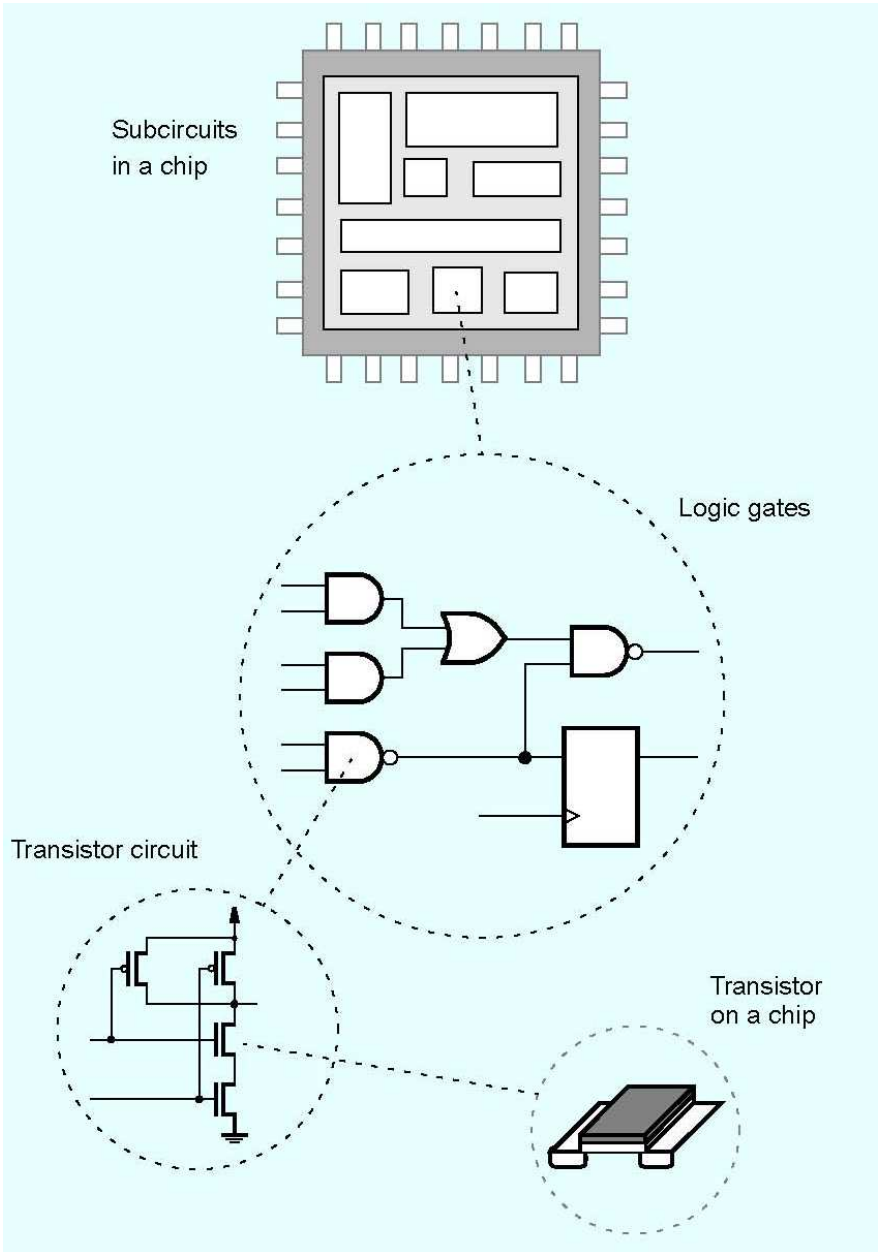


Figure: 7-segment Display

Digital Hardware System



Digital Hardware System - II



Printed Circuit Board