CDA 3120 Digital Networks
Spring 1998

- Room: 103 Love Building

- Time: 12:20 PM – 1:10 PM MWF
Instructor: K. A. Gallivan

- 214 Love Building
- 644-6410,
- gallivan@cs.fsu.edu
- Office Hours:
  - Tuesday 10:30 AM – 12:00 PM
  - by appointment at other times

Teaching Assistant: Rebecca Cao

- 102 C Caruthers Hall
- 644-0050
- cao@cs.fsu.edu.
- Office hours will be announced.
Class Materials


- Notes: Postscript files of the lecture notes, announcements, homeworks, and solutions will be available via anonymous ftp. Check this directory regularly.

  - machine: ftp.cs.fsu.edu
  - directory pub/gallivan/CDA-3102
  - login: anonymous, password: full email address
  - mode: binary mode of ftp transfer is recommended.
  - compression: The files will be compressed using the Unix gzip utility.
  - printing: It is recommended that you print more than one note page on a single sheet of paper, e.g., four note pages on each side of a sheet is still readable.
Exams:

- 2 exams during the semester at approximately 1/3 and 2/3 of the way through
- a **comprehensive** final
- All exams are open textbook and open class notes. No other material is allowed.

Grading:

- maximum of two semester exams - 30%
- homework - 25%
- lab - 10%
- final exam - 35%

Lab:

- details will be announced next week
Homework

- Homework will be assigned on approximately a weekly basis.

- You will receive credit for a homework assignment if:
  - it is turned in by the end of the day it is due (5 PM)
  - and if it is evident that a reasonable attempt at solving all of the problems was made.

- Your credit for the homework does not depend on the correctness of your solutions. It does depend on making a concerted effort to solve all of the problems.

- You may elect to not turn in up to 2 assignments and still receive full homework credit for the semester.

- It will be very rare that missing any additional homework without losing full credit will be allowed.

- Detailed solutions will be provided to each homework assignment.
Goals:

- To provide an introduction to the analysis and synthesis of the basic components of digital computers at the logic design level.

- To provide the background in Boolean algebra and switching theory necessary to understand more advanced techniques in VLSI CAD systems
Two main considerations when building systems:

- digital vs. analog
- physics of representation:
  - mechanical
  - electrical
  - acoustic
  - EM – infrared, video, radio

Analog devices specify information via continuous functions.

Operations on the information exploit similarity of physics of representation to operation desired.

Digital devices encode information via a finite number of states.

Operations are used to model desired behavior.

Analog systems tend to be special purpose and often very high-speed while digital systems are much more flexible due to programmability.
Examples

- mechanical, analog device – spring driven clock
- mechanical, digital device – key (binary input) and lock (binary output)
- mechanical, digital device – mechanical calculator, multistate digital I/O, (this ignores the analog I/O between human and device)
- optical, analog device – lens
Analog devices often use multiple physical representation of information.

The choice is made based on the type of operations to be performed on the data at that point in the system

- night vision goggles: infrared $\rightarrow$ electrical $\rightarrow$ video

- telephone: acoustic $\rightarrow$ mechanical $\rightarrow$ electrical $\rightarrow$ mechanical $\rightarrow$ acoustic
In practice, most useful systems are hybrids of digital and analog. Example: digital cellular telephone.

- voice to voltage/current: acoustic → mechanical → electrical (analog)

- conversion to digital electrical followed by various digital signal processing – noise suppression, echo cancellation

- transmission requires converting time sequence of digital information to radio frequency analog information – this is not the same as the radio analog information found in non-digital phones

- reception and display reverses the process.

- analog-to-digital conversion - sampling, quantization error concern

- digital-to-analog conversion - interpolation to reconstruct continuous function; in the case of encoding digital information, discontinuous input causes difficulties
Electrical Networks

- The basic components are analog since their function is based on the time behavior of current and voltage.

- To implement digital electrical systems, the continuous waves must be made to look like square waves therefore emulating two-state logic (a switch).

- The control of this approximation is crucial to reliable devices:
  - voltage thresholds must be set to correspond to 0 and 1 states
  - rise and fall times must be short enough to cause efficient operation but long enough to be reliable
  - jitter around nominal voltage values for 0 and 1 state must be controlled.
• The design and behavioral modelling of these devices requires integration of ordinary and partial differential equations mathematics.

• The basic components are used to form digital cells that implement basic gate and switch functions.

• They have distinct I/O connections and area boundaries that allow them to be placed on a chip in relation to other cells.
<table>
<thead>
<tr>
<th>Level</th>
<th>Math</th>
<th>Components</th>
<th>Physical Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transistor</td>
<td>differential equations, i/v curves</td>
<td>transistors, resistors, capacitors</td>
<td>analog, digital cells</td>
</tr>
<tr>
<td>Gate, Switch</td>
<td>Boolean algebra, FSM’s</td>
<td>gates, latches flip-flops</td>
<td>functional units</td>
</tr>
<tr>
<td>Register</td>
<td>instructions, generalized FSM’s</td>
<td>adders, registers, counters</td>
<td>chips</td>
</tr>
<tr>
<td>Processor</td>
<td>programs</td>
<td>processors, ASIC’s, memories</td>
<td>boards, multiple chips</td>
</tr>
</tbody>
</table>
• We will consider in detail the gate level mathematics and design.

• Options for the form of the components that are used as building blocks will be considered.

• Optional forms of the components created for use at the register level will be considered.

• Where the single chip boundary falls is changing rapidly.

• Hybrid analog/digital systems on a single chip are hot topic for technology and application reasons.
Overview of Course

- Information representations:
  - various encodings of digital information
  - choice depends upon the operations to be performed and the hardware available
  - error detection/correction possible

- Boolean algebra and functions
  - switches, gates, networks, and the technology options for their implementation
  - complete gate sets
  - options for Boolean function specification
  - basic properties and theorems
  - limits of theory
• Combinational logic – memory-less operations

• Analysis of combinational devices
  – given a device determine function (assuming correct operation)
  – given a device and the function it should implement determine if there are any errors of particular types

• Synthesis of combinational devices
  – Textual description of function → mathematical form
  – Create fundamental characterizations of Boolean function(s) implemented
  – Create optimal (or near-optimal) realization in a given technology.

• We will consider both single-output and multiple-output devices.

• Combinational components
  – Consider the designs of basic register level components found in digital computers
- Sequential logic – memory-based functions

- Alterations of combinational networks to store information
  - feedback loops
  - latches

- fundamental mode (asynchronous) vs. pulse mode (synchronous) operation

- synchronous memory devices
  - flip-flops from latches

- Analysis of sequential devices
  - finite state machines (FSM)
  - Moore and Mealy machines vs. practice
• Synthesis of synchronous sequential machines
  – Textual description of function → FSM
  – State reduction
  – State assignment
  – Realization in terms of selected memory devices

• Important sequential components of a digital computer
  – registers
  – register files
  – counters
  – random-access memories