Welcome to 6.191 [6.004]!

Computation Structures

Spring 2023
6.191 Course Staff

Instructors

Song Han
songhan@mit.edu

Silvina Hanono Wachman
silvina@mit.edu

Daniel Sanchez
sanchez@csail.csail.edu

Teaching Assistants

Keshav Gupta
Min Thet Khine
Alexandra Li
Georgia Shay
Krittamate Tiankanon
Travis Ziegler
Computing Devices Then...

ENIAC, 1943  30 tons, 200KW, ~1000 ops/sec
Computing Devices Now

Typical 2023 laptop
1kg, 10W, 1 trillion ops/s
An Introduction to the Digital World

- Computer programs
- Virtual machines

- Computer systems
  - Operating systems, virtual memory, I/O
  - Instruction set + memory

- Computer architecture
  - Processors, caches, pipelining
  - Digital circuits

- Digital design
  - Combinational and sequential circuits
  - Bits, Logic gates

- Devices
- Materials
- Atoms
The Power of Engineering Abstractions

Good abstractions let us reason about behavior while shielding us from the details of the implementation.

*Corollary:* Implementation technologies can evolve while preserving the engineering investment at other levels.

Leads to hierarchical design:
- Limited complexity at each level ⇒ shorten design time, easier to verify
- Reusable building blocks

Virtual machines

Instruction set + memory

Digital circuits

Bits, Logic gates
Course Outline

- **Module 1: Digital design**
  - Combinational and sequential circuits

- **Module 2: Computer architecture**
  - Simple and pipelined processors
  - Caches and the memory hierarchy

- **Module 3: Computer systems**
  - Operating system and virtual memory
  - Parallelism and synchronization
6.191 Builds On 6.190

- 6.191 assumes knowledge of C programming and RISC-V assembly as taught in 6.190

- Starting this term, **6.190 is a hard pre/co-requisite**
  - You can’t take 6.191 without 6.190
  - For Spring 2023 only, we allow clearing this prereq with 6.08 (which taught C, but not assembly), but we strongly discourage it

- Several exciting changes this term!
  - More focus on computer systems (OS, parallelism)
  - New labs and design project
Our Focus: Systems with General-Purpose Processors

- General-purpose processors are the basic building block of computer systems
  - Understanding them is crucial even if you do not plan to work as a hardware designer
- General-purpose processors are the most sophisticated digital systems that exist today
  - Understanding them will help you design all kinds of hardware

By the end of the term you will have designed a simple processor from scratch!
Course Mechanics

- 2 lectures/week: slides, videos, and reference materials on website
- 2 recitations/week: work through problems using skills and concepts from previous lecture
- 7 mandatory lab exercises
  - Online submission + in-person check-off meetings
  - Due throughout the term (7 free late days meant to give you flexibility, cover short illnesses, etc.; see website)
  - **Lab exercises are to be completed individually – must abide by collaboration policy**
- One optional open-ended design project
  - Due at the end of the term
- 3 quizzes: Mar 9, Apr 20, May 11 (7:30-9:30pm)
  - If you have a conflict, contact us to schedule a makeup
Recitations

- 8 recitation sections on Wed & Fri
  - Select your recitation section on the website
- Recitations
  - Solve tutorial problems with instructor
  - Rely on having attended lecture and attempting the post-lecture problems prior to each recitation
- Recitation participation is worth 5% of your grade
  - Must attend and complete post lecture exercises for credit
  - Lecture exercises due by 10am
  - Everyone has 4 excused absences
- May opt out of recitation participation by March 10
  - If you opt out, that 5% of grade will come from quizzes
  - We believe recitations are the right choice for most of you
Grading

- 80 points from labs,
- 20 points from design project,
- 90 points from quizzes,
- 10 points from recitation participation

Fixed grade cutoffs:
- A: Points $\geq 165$
- B: Points $\geq 145$
- C: Points $\geq 125$
- F: Points $< 125$ or not all labs complete or quiz avg $< 40\%$
Resources

- The course website has up-to-date information, handouts, and references to supplemental reading: http://6191.mit.edu

- We use Piazza extensively
  - Fastest way to get your questions answered
  - All course announcements are made on Piazza

- We will hold regular office hours to help you with labs, infrastructure, and any other questions
  - Office hours are in 32-083
  - Schedule and lab queue are available on the website
  - If you must quarantine, let us know and we can accommodate virtual checkoffs
Lab Setup and Bash Tutorials

- Tuesday Feb 7 – 32-083 – 7-8pm
- Wednesday Feb 8 – 32-083 – 7-8pm
The Digital Abstraction

Building Digital Systems in an Analog World
Analog vs. Digital Systems

- Analog systems represent and process information using **continuous signals**
  - e.g., voltage, current, temperature, pressure, ...

- Digital systems represent and process information using **discrete symbols**
  - Typically binary symbols (bits)
  - Encoded using ranges of a physical quantity (e.g., voltage)

Digital systems tolerate noise
Example: Analog Audio Equalizer

Input: Voltage signal representing sound pressure

Expected output: Frequency-equalized voltage signal
Example: Analog Audio Equalizer

Input: Voltage signal representing sound pressure

Expected output: Frequency-equalized voltage signal

Does output match expected output? Not quite!

Why or why not? Noise
Manufacturing variations
Components degrade over time

February 7, 2023
MIT 6.191 Spring 2023
Keep in mind that the world is not digital, we would simply like to engineer it to behave that way. In the end we must use real physical phenomena to implement digital designs!
Using Voltages “Digitally”

• Key idea: Encode two symbols, “0” and “1” (1 bit)
• Use the same convention for every component and wire in our digital system

Attempt #1:

\[ V < V_{TH} \]
interpreted as “0”

\[ V \geq V_{TH} \]
interpreted as “1”

Not quite correct. Why? Hard to distinguish \( V_{TH}-\epsilon \) from \( V_{TH}+\epsilon \)

Attempt #2:

\[ V \leq V_L \]
interpreted as “0”

\[ V_L < V < V_H \]
“Undefined”

\[ V \geq V_H \]
interpreted as “1”

✓ ?
Will This System Work?

Upstream device transmits a signal at $V_L - \varepsilon$, a valid “0”. Noise on the wire causes the downstream device to receive $V_L + \varepsilon$, which is undefined.

How can we address this?

*Output voltages should use narrower ranges, so that signal will still be valid when it reaches an input even if there is noise.*
Noise Margins

Proposed fix: Different specifications for inputs and outputs

- Digital output: “0” ≤ \( V_{OL} \), “1” ≥ \( V_{OH} \)
- Digital input: “0” ≤ \( V_{IL} \), “1” ≥ \( V_{IH} \)
- \( V_{OL} < V_{IL} < V_{IH} < V_{OH} \)

A digital device accepts marginal inputs and provides unquestionable outputs (to leave room for noise)
Digital Systems Fight Noise

Analog systems: Noise accumulates

\[ V_{\text{I}} \xrightarrow{\varepsilon_1} V_{\text{I}} + \varepsilon_1 \xrightarrow{f} f(V_{\text{I}} + \varepsilon_1) \xrightarrow{\varepsilon_2} f(V_{\text{I}} + \varepsilon_1) + \varepsilon_2 \xrightarrow{g} g(f(V_{\text{I}} + \varepsilon_1) + \varepsilon_2) \]

Digital systems: Noise is canceled at each stage

\[ V_{\text{I}} \xrightarrow{\varepsilon_1} V_{\text{I}} + \varepsilon_1 \xrightarrow{f} f(V_{\text{I}}) \xrightarrow{\varepsilon_2} f(V_{\text{I}}) + \varepsilon_2 \xrightarrow{g} g(f(V_{\text{I}})) \]

Intuitively, canceling noise requires *active components*, i.e., components that inject energy into the system.
**Buffer**: A simple digital device that copies its input value to its output

**Voltage Transfer Characteristic (VTC)**: Plot of $V_{out}$ vs. $V_{in}$ where each measurement is taken after any transients have died out.

VTC must avoid the shaded regions (aka “forbidden zones”), which correspond to valid inputs but invalid outputs.

Note: VTC does not tell you anything about how fast a device is — it measures static behavior, not dynamic behavior.
Voltage Transfer Characteristic

1) Note the center white region is taller than it is wide (\(V_{OH} - V_{OL} > V_{IH} - V_{IL}\)). Net result: device must have GAIN > 1 and thus be ACTIVE.

2) Note the VTC can do anything when \(V_{IL} < V_{IN} < V_{IH}\).
Types of Digital Circuits

- **Combinational circuits**
  - Do not have memory
  - Each output is a function of current input values
  - Examples:
    - Inverter
      - 0 → 1
      - 1 → 0
    - AND
      - Output is 1 if both inputs are 1, 0 otherwise

- **Sequential circuits**
  - Have memory, i.e., *state*
  - Each output depends on current state + current inputs
Summary

- Digital systems tolerate noise
- Digital encoding
  - Valid voltage levels for representing “0” and “1”
  - Undefined range avoids mistaking “0” for “1” and vice versa
  - Noise margins require tougher standards for outputs than for inputs

- Complete post-lecture questions by 10am tomorrow
  - 6191.mit.edu > Material > Lectures > L1 questions
Thank you!

Next lecture:  
Combinational Devices and Boolean Algebra