Zachary Yedidia

zyedidia@college.harvard.edu

zyedidia.github.io, github.com/zyedidia

Education

Harvard College — Cambridge, MA

A.B./S.M. Candidate in Computer Science — *Aug. 2017 – May 2021*

Selected coursework:

- Systems Security (CS263)
- Computational Linguistics and NLP (CS187)
- Senior Thesis Research (CS91r)
- Research Topics in Computer Architecture (CS247r)
- Big Data Systems (CS265)
- Introduction to Semantics (Ling106)
- Advanced Computer Architecture (CS246)
- Special Topics in Edge Computing: Autonomous Vehicles (CS249r)
- Probabilistic Analysis and Algorithms (CS223)
- Data Systems (CS165)
- Data Structures and Algorithms (CS124)
- Computational Neuroscience (MCB131)
- Electromagnetism and Statistical Physics (Physics15b)
- Compilers (CS153)
- Theory of Computation (CS121)
- Optimization: Methods and Models (AM121)
- Circuits, Devices, and Transduction (ES152)
- Operating Systems (CS161)
- Design of VLSI Circuits and Systems (CS148)
- Discrete Mathematics (CS20)
- Systems Programming and Machine Organization (CS61)
- Computing Hardware (CS141)
- Scientific Computing (AM111)
- Mathematical logic (Phil140)
- Linear Algebra and Differential Equations (Math21b)

Concord Academy — Concord, MA

High school — *Sept. 2013 – June 2017*

- **Teaching**: I taught an elective in the Computer Science department called *Game Programming with Go* under the supervision of the CS teacher. The class materials can be found at https://github.com/CAGameProg.

Selected Course Projects

- *WFilter* (Systems Security, Fall 2020): A customizable sandboxing and tracing tool. WFilter allows a user to attach small Webassembly programs to various events (such as system calls) in a child process. WFilter will run the WASM programs when the events happen, and allow the WASM programs to reject or modify the system calls, allowing sandboxes or advanced tracers to be expressed easily without worrying about the details of `ptrace`. Deep pointer inspection is supported along with many other features that are not supported by the somewhat similar `seccomp` interface.
- *Quantized Multi-Columnar Predicate Evaluation* (Big Data Systems, Spring 2020): Provided more expressive predicates for high-performance scans on quantized data over multiple columns. Predicates can be efficiently stored in a lookup table which can be accessed by SIMD instructions during the scan.
- *Near-Memory Processing for DBs* (Research Topics in Computer Architecture, Spring 2020): Continued work on a project for extracting columns from a row-store directly in the disk controller, which avoids sending wasted row data up the memory hierarchy when reading columns.
- *Precision Batching for Quantized Matrix Multiplication* (Research Topics in Edge Computing, Fall 2019): Precision Batching applies bit-serial quantized matrix multiplication to GPU architectures for performance gains over NVIDIA Cutlass.
- *A Survey of Adaptive AMQs* (Probabilistic Analysis and Algorithms, Fall 2019): Analyzed and summarized the algorithms for Broom Filters (a version of bloom filters that is adaptive) and Adaptive Cuckoo Filters.

Publications and Writing

Maximilian Lam, Zachary Yedidia, Colby Banbury, Vijay Janapa Reddi. "Quantized Neural Network Inference with Precision Batching" (2020). In submission to MLSys. Link.

Zachary Yedidia, "SystemVerilog Guide" (2020). Used in course materials for CS 141 (Spring 2020) at Harvard. Link.

Open Source Projects

Micro Text Editor Website, GitHub Project

I created and launched a text editor called Micro in April 2016. Micro is a Go project with over 15,000 stars on GitHub, more than 500,000 downloads, and 100+ contributors. It aims to be a successor to Nano as a simple to use terminal-based text editor. Micro was the subject of multiple news articles and has been featured on the front page of Hacker news multiple times. Micro is available in many package managers such as: Homebrew, Apt (for Ubuntu Focal and Debian Buster), Snap, AUR, Chocolatey and more.

GPeg GitHub Project

In-progress library for PEG parsing, as part of my senior thesis research with Professor Stephen Chong. GPeg uses a parsing virtual machine for dynamic parser generation, and implements a novel algorithm for efficient incremental parsing.

Perforator GitHub Project

Perforator is a tool for recording Linux "perf" metrics like cache misses, branch mispredictions, CPU cycles, etc... for certain parts of a program like functions or source code regions (as opposed to `perf stat` which only records over entire program lifetimes). It works by using `ptrace` and inserting software breakpoints to enable and disable profiling (Perforator reads the ELF symbol table and possibly DWARF debugging information to determine where to place breakpoints). Perforator supports position-independent ELF executables and multithreaded programs (with limitations).

Literate Programming Tool Website, GitHub Project

A tool for compiling Literate programs written in any programming language. Featured on the front page of Hacker News in September 2015. The article "Write your Own Virtual Machine" was written using Literate. Link.

SFML.jl, Chipmunk.jl Talk, GitHub Project

Graphics and physics libraries presented at JuliaCon 2015 at MIT.

Experience

Harvard University Cambridge, MA

HCRP Research Fellow (Advisor: Prof. Stratos Idreos) *Summer 2020*

- Worked on a project for improving hash function performance for use in database hashtables and filters, implementing the novel technique in state-of-the-art hashtables/filters for benchmarking and analysis.

Raytheon Company Tucson, AZ

Internal Research and Development Intern *July – August Summer 2019*

Advanced Missile Systems

- Worked on a research project in DARPA's Electronics Resurgence Initiative (ERI).

Princeton University Princeton, NJ

Research Assistant (Advisor: Prof. Naveen Verma) *May – June Summer 2019*

- Wrote software for a state-of-the-art in-memory computing ASIC using mixed-signal SRAM technology.
- Mapped applications to the hardware including signal processing and machine learning applications.
- Characterized and benchmarked performance and noise.

Harvard University Cambridge, MA

PRISE Research Fellow (Advisor: Prof. Eddie Kohler) *Summer 2018*

- Worked on the C++ transactional memory system (called STO) developed by Professor Eddie Kohler's computer systems research group, implementing and benchmarking a transactional Adaptive Radix Tree that outperformed the existing transactional Masstree.

Cogito Health Boston, MA

Software Engineering Intern *Summer 2014*

- Worked on the development of a large voice-recognition and analysis program written in Java, as well as prototypes for various android apps.

Teaching

Systems Programming and Machine Organization (CS61)

Teaching Fellow *Fall 2020*

Course heads: Prof. Eddie Kohler, Prof. Minlan Yu

- Held office hours and lecture viewings for the course, which was offered in an all-virtual format.
- Teaching evaluations: 4.9/5.0.

Computing Hardware (CS141)

Teaching Fellow *Spring 2019, Spring 2020*

Course heads: Prof. David Brooks, Prof. Vijay Reddi

- Helped to create, distribute, and organize the programming component of CS141, introducing students to SystemVerilog, digital design, and computer architecture using FPGAs.
- Created many new materials for the class including guides on combinational and sequential logic, FSM design, and caching. Upgraded development to use SystemVerilog and Xilinx Vivado with Digilent Nexys A7 FPGA boards.
- Recruited and organized teaching staff for the course.
- Teaching evaluations: 4.8/5.0, Derek Bok Center teaching award (2019). No teaching evaluations in 2020 due to COVID.

Compilers (CS153)

Teaching Fellow *Fall 2019*

Course head: Prof. Stephen Chong

- Held office hours to help students with the class projects which involved building an optimizing compiler targeting LLVM.
- Teaching evaluations: 5.0/5.0.

Programming Skills

Primary Interests:	Computer Systems, Hardware/Architecture, Compilers.
Languages:	C/C++, Go, SystemVerilog, Python, Java, D, OCaml, Matlab, Julia, Lua, Perl.
Tools:	Vim, Git, Xilinx Vivado, PyTorch, LaTeX.