

Rahul Chaudhry

Home Address

Mobile: (650) 796 - 5908
rahul@rahulc.com
<http://rahulc.com/>

Work Address

Phone: (503) 685 - 0667
rahul_chaudhry@mentor.com
<http://mentor.com/>

Objective To work as a software developer in a technically strong and result oriented group so as to utilize my skills and experience to a maximum extent.

Work Authorization Citizenship: **India**
US Visa: **H-1B** (Alien in a specialty occupation)

Education

- **Master of Science (M.S.)** in Computer Science (May 2004)
Department of Computer Science,
Yale University, USA.
GPA: 3.8/4.0
- **Bachelor of Technology (B.Tech.)** in Computer Science and Engineering (May 1999)
Department of Computer Science and Engineering,
Indian Institute of Technology (IIT) Kharagpur, INDIA.
GPA: 8.50/10.00

Interests Programming Language Design, Compilers, Interpreters, Simulators, Code Generators.

Experience

STAFF ENGINEER **Mentor Graphics, USA**
August 2008 – Present
SR. S/W DEV. ENGINEER
August 2005 – July 2008

SR. R&D ENGINEER **Jasper Design Automation, USA**
February 2005 – August 2005
R&D ENGINEER
June 2004 – January 2005

GRADUATE STUDENT **Yale University, USA**
August 2002 – May 2004
Teaching Fellow: *CS112 - Introduction to Programming* with Java (Spring 2004).

Graduate Level Courses Taken:

Compilers and Interpreters,	Formal Semantics,
Operating Systems,	Theoretical Methods in Computer Science,
Computer Networks,	Mobile Computing and Wireless Networks,
Artificial Intelligence,	Intelligent Robotics,
Parallel Programming Techniques,	Numerical Computation.

RESEARCH CONSULTANT **IIT Kharagpur, INDIA**
June 2000 – July 2002

- Designed and implemented a processor description language.
- Automatic generation of cycle accurate simulator and assembler for a processor.
- Integration of *CR16c* on FPGA board with *gdb* for remote debugging.
- *Magic-C* to *SystemC* translator design and implementation.

SOFTWARE ENGINEER **Cadence Design Systems, INDIA**
June 1999 – June 2000

- Developed STAMP parser, enhanced and maintained TLF and TLB database libraries.
- Wrote a 'C++ class header files' to 'C struct definitions' translator to ease library code debugging and maintenance.

Received the **Institute Prize for the Best Project Work** for my work on the design and implementation of *Unicorn: A multi-paradigm imperative/logic programming language*.

Relevant Course-work:

Programming and Data Structures,	Design and Analysis of Algorithms,
Computer Organization and Architecture,	Switching Theory and Logic Design,
Operating Systems,	Formal Languages and Automata Theory,
Compiler Design,	Computational Complexity,
Computer Networks,	Parallel Algorithms,
Computer Graphics,	Computational Geometry,
Systems Programming,	Database Management Systems,
Software Engineering,	Discrete Structures.

Language Design

- Designed and implemented *Unicorn*. The basic idea was to add new control constructs to *C* that would allow backtracking based *Prolog* like goal searching control flow within and across *C* functions, and provide *C++* like classes and overloaded operators to enable the programmer to define unification as just a class library.
- Designed and implemented a processor description language for automatic generation of cycle accurate simulator, assembler and disassembler for a processor from a succinct definition of its instruction set architecture, execution model and memory model.

High-Level Code Generators

- The implementation of *Unicorn* generates equivalent *C* code for an input program.
- The implementation of the processor description language generates *C++* code files for the simulator, assembler and disassembler of the processor.
- Wrote a translator to convert a high level *Magic-C* system description to equivalent *SystemC* code.
- Wrote a translator to convert a high level *Magic-C* system description to equivalent *C++* code for a time-traveling simulator for the system, that allowed going back in simulation as well as forward.

Compilers

- Wrote a compiler for a reasonably large subset of *C*, using *lex* and *yacc*.
- Wrote a compiler in *ML* for a *Pascal* like language featuring garbage collection, using *ml-lex* and *ml-yacc*.

Interpreters

- Wrote an interpreter for the *Lambda Calculus*.
- Wrote a semantic interpreter, in *Haskell*, for a simple hybrid imperative/functional programming language.

Simulators

- Wrote an event based simulator for digital circuits. Implemented a graphical user interface for editing and simulating digital circuits.
- Wrote a cycle based simulator for *REX* networks. Implemented and tested a one instruction set computer (a *mov-machine*) using *REX* network components.
- Wrote a cycle accurate simulator for National Semiconductor's *CR16c* processor.

Languages & Skills

Programming in *C*, *C++*, *ML*, *Haskell*, *Prolog*, *Java*, *Scheme*, *MIPS/x86 Assembly*.
Tools: *Lex*, *Yacc*, *Gdb*, *Bash*, *Vim*, *RCS*, *SVN*, *GIT*, *L^AT_EX*, *HTML*.

Awards

- **Institute Prize for the Best Project Work** for *Unicorn*.
- **Yale University Fellowship** covering tuition and expenses for study at Yale.

Miscellaneous

- **AGRE Score:** 99 percentile in Computer Science Subject Test.
- **GRE Score:** Analytical 800, Quantitative 800, Verbal 650 — Total 2250/2400.

References

- Available upon request -