**MALCOLM A. MUMME**

P.O. Box 85062, Tucson, AZ, 85754-4004

(951) 808-5580

<http://www.cs.ucr.edu/~mummem/>

mamumme@acm.org

**PROFILE**

* Software engineer since 1980, with experience in simulation, embedded real-time systems, microcoding, DSP, processor design, systems programming, theory.
* Many programming languages, applications and operating systems.
* Committed to high technical standards. • Likes math, physics, parallel processing.
* Can work autonomously, independently or in teams. • Self-motivated.

**PARTIAL COMPUTER SKILLS**

1. Advanced data structures/algorithms
2. Ada
3. Assembly
4. C / C++ / C++11
5. Haskell
6. Java
7. Pascal
8. VHDL

 **EDUCATION (1999-2017)**

University of California at Riverside, Riverside, California (UCR)

PhD in Computer Science (specialized in formal methods/verification) (Expected) 2020

Master of Science degree in Computer Science (MSCS) December 2008

Bachelor of Science degree in Computer Science (BSCS) June 2004

 **INDUSTRIAL EXPERIENCE (and other projects)** **(1980-1998)**

**Software Engineer, Hughes Aircraft/Raytheon Systems Company, El Segundo, CA**

•Designed and coded functional improvements to radar display simulator.

•Simplified HW design for new target echo simulation processor, increasing flexibility.

•Debugged and patched legacy microcode (128 bit wide, 7 functional units, sub-clock timing), sometimes using oscilloscopes, and created microcode to C translator.

•Designed/coded radar clutter simulator output quality improvements.

•Designed API for open simulation interface, supporting reversible simulation.

•Coded cellular processor simulator updates/improvements, including state save/restore.

•Wrote Haskell program to generate simulator code, preventing many potential errors.

•Found errors inspecting cellular processor logic schematics, saving designer debug time.

•Designed/built custom *Apple][e* memory card for MX (see below) (personal project).

•Design/code/integrate/tested real-time interrupt driven embedded C/assembly program.

•Design/code/studied timing for image/DSP algorithm cores for three processor designs.

•Wrote CGI code (ray tracing, radiosity) on *Apple][e* in assembly/Pascal. (personal projects).

•Designed/coded data flow analysis unit for DSP array memory allocation program.

•Designed/coded combinator reduction-based macro expander MX (personal project).

•Mission reliability prediction program coding/GUI/database.

•Customized/maintained/enhanced/debugged/delivered/installed incremental assembler.

•Debugged and enhanced real-time radar test system control code.

**RESEARCH PROJECTS/TEACHING EXPERIENCE**

2018-20 Implementing new VLDD library per ‘future work’ section of dissertation.

2017-18 Adjunct at calbaptist.edu: CSC522 (Agile Meth.), CSC524 & EGR424 (RoR Web Apps), CSC526 & EGR326 (S/W Des. & Arch.), EGR506 (Rsrch Meth.).

2015-17 Wrote **517**-pg. dissertation on Variable-Labeled Decision Diagrams (VLDDs).

2015 Completed proposed VLDD research. Wrote **10000**+ lines C++ code in 5 months.

2014 Wrote **68**-page dissertation proposal. Taught lab for CS008: Intro to computing.

2013 Proved VLDDs canonical. Graded for CS246: Advanced Verification Techniques.

* Taught lab for CS010: Intro to programming. Switched topic after P1 defense fail.

2012 Wrote first dissertation proposal (P1) (**83** pages). Invented VLDD prototype.

2011 Wrote paper [2] as first author, with advisor as co-author.

* Re-coded saturation-based symbolic algorithm for weak and strong bisimulation.
* Taught lab for CS012: Intro to Programming II, and for CS008.

2010 Optimized code, adapted models, and wrote paper [1] as first author, with advisor.

- TA for CS061: Machine Organization and assembly language programming, for CS008, and for CS181: Principles of Programming Languages):

* Designed saturation-based symbolic algorithm for weak and strong bisimulation.

2009 Implemented novel saturation-based symbolic algorithm for strong bisimulation of systems having only deterministic transitions. Studied algorithm performance

* Taught lab for CS180: Introduction to Software Engineering

2008 Finished MS in CS with project comparing bisimulation partition representations.

* Found about 15 bugs in C-language -> VHDL translator.

2007 Implemented sparse matrix multiply in C dialect for reconfigurable computing.

* Designed (for VHDL implementation) highly super-scalar sparse matrix multiply.
* TA for CS179E: Senior Project in Computer Science: Compilers.
* Designed optimal grid drawing method for onion graphs of 3 or 4 vertices/layer.
* Devised effective variable ordering heuristic for compact decision diagrams.

2006 Taught lab for CS152: Compiler Design and for CS008.

* Designed/coded virtual blackboard for drawing with dynamic Lorenz transforms.

2005 Taught lab for CS008: Introduction to computing. Begin grad school classes.

2003 Designed/implemented programming language for ultra-fine-grained parallelism.

2002 Wrote multi-player distributed flight simulator game in Java with Java graphics

* Determined all plausible 1-D physics relativistic energy-momentum transforms.

**PAPERS**

**[1]** A fully symbolic bisimulation algorithm. In: Delzanno, G., Potapov, I. (eds.)

Proc. 5th Intl. Workshop on Reachability Problems. pp. 218–230. Springer-Verlag

**[2]** An efficient fully symbolic bisimulation algorithm for nondeterministic systems. In:

International Journal of Foundations of Computer Science, V. 24, Iss. 2, pp. 263-282.

**PATENTS**

* #5,379,444: “simplified synchronous mesh processor and array”
1. A processor design that is nearly the most extremely minimal case of a SIMD array.
* #5,815,728: “processor array” (link: <http://www.google.com/patents/US5815728>).
1. A scheme for improving I/O pin usage of certain 2D mesh-connected processor chips.