Binu K. Mathew

bmathew@gmail.com (650) 283-9142
http://www.cs.utah.edu/~mbinu/

Research Interests

Computer architecture, low power processors and embedded systems, power estimation and modeling, perception, media and streaming architectures, VLSI design, compiler and CAD algorithms, operating systems and networking

Education

Post Doctoral Research, Stanford University, CA **April 2005**
Topic: *Stream Processors*
Advisor: Prof. Bill Dally

Ph.D. in Computer Science, University of Utah, Salt Lake City, UT **August 2004**
Dissertation: *The Perception Processor*
Advisor: Prof. Al Davis

MS in Computer Science, University of Utah, Salt Lake City, UT **January 2000**
Thesis: *Parallel Vector Access: A Technique for Improving Memory System Performance*
Thesis Advisors: Prof. Al Davis, Prof. Sally A. McKee

B. Tech in Computer Science, University of Kerala, India **October 1995**
Thesis: *Design and Implementation of a Micro-kernel Operating System*
Thesis Advisor: Prof. Frahad Musadeekh

Honors

- Winner, 2004 Siemens TTB International post-doctoral entrepreneurship contest.
- Finalist: 1995 National Talent Search in Software Engineering, India.

Professional Experience

Founder of Satva Design Automation / Innovator (Post-doctoral Researcher), Siemens Technology-To-Business Center, Berkeley
April 2005-Present

In April 2005, I started an EDA company named Satva Design Automation with Siemens as the angel investor. **(`http://www.satvad.com`)**. I lead a three member research and development group investigating rapid ASIC and SoC design using customized stream processors. We are currently experiencing difficulties on settling equity/ownership with Siemens and I am looking for another position. My current role involves research and development in processor architecture, optimizing parametrized compilers, automated design exploration, hardware generation etc. I am also responsible for project management, business plans, negotiating with external vendors and investors, contracts, incorporation, etc.

Development tasks included analysis of embedded applications, design of application specific processors, compilers and tool chains, integrating design exploration and heuristic optimization with Verilog interpreters, GUIs, code generators etc. I had to deal with pretty much every aspect of computer science in this project from writing parsers, genetic algorithms, industrial automation code etc. to XML and bug fixes contributed for the Firefox/Mozilla web browser.

As a part of this project we used our stream processor technology to prototype a PLC processor design. Under my direction, my three member team has designed and implemented a custom RISC processor for industrial automation in a $90nm$ CMOS process and developed a compiler that performs binary translation from the legacy instruction set

to the new instruction set. We have implemented from scratch everything from the front-end that lexes and parses the STL language to dominators, dataflow analysis, SSA conversion, graph-coloring register allocation, optimization passes etc. Siemens A&D in Germany is the world leader in PLC processors used for industrial automation. Our PLC design if carried forward will become the heart of a PLC product line with annual sales exceeding $1.5 billion.

Post-doctoral Scholar, CVA Group, Stanford University
July 2004-April 2005
Research in Stream Processing, Scheduling and on-chip memory allocation, Energy-delay optimization of low-power media processors. Design of an inverse square root floating point unit for the Merrimac Streaming Supercomputer. Development work on the simulator for the Imagine/Merrimac stream processors. Simulation studies and simulator development to compare IBM Cell, MIT RAW and Stanford Merrimac/Imagine. Automated floor plan generator for stream processors. Mathematical work on the theory of space and time scheduling. Job responsibilities also included writing research grants and providing consulting for a research group of 12 doctoral students.

Research Assistant, Architecture Group, School of Computing, University of Utah
August 1997-July 2004
Processor design:

Designed and implemented a power efficient VLIW processor for a variety of perception, and streaming tasks in 0.25u and 0.13u CMOS processes. Developed tools for the automatic generation of domain-specific stream processors that can accelerate the performance of a variety of perception algorithms at low power budgets. Developed a compiler to transform GCC compiler intermediate code to micro-code. Designed a synthesizable low-power MIPS processor similar to the R4600. Ported Linux to this processor, modified GCC and GNU binutils. Performance analysis of speech recognition and computer vision applications. Originated the concept and contributed to the development of a face recognition system for video.

Memory controller design: (Impulse Adaptive Memory Controller Project for DARPA/Airforce Research Labs) Micro architecture and VLSI design for an adaptive high performance memory controller for a 0.25u process. Development of functional and trace driven simulators.

Operating Systems: Ported Linux Ethernet drivers (Intel Ether Express Pro, DEC Tulip) to the Flux OS kit. Created glue code to use the Linux network drivers along with the pre-existing Free BSD TCP/IP stack in the Flux OSkit. Implemented some security features required to permit using user space message buffers in the Fluke micro-kernel's messaging API.

Graduate Technical Intern, Apple Computer, Inc., Cupertino, CA
Summer 2001-Fall 2001
Design and implementation of a floating point unit for multimedia in a deep sub-micron process. I reverse engineered the Multiply-add floating point unit in the Altivec SIMD portion of the IBM PowerPC. Apple needed a clean room implementation of this circuit in a 0.13u process. I wrote test code to analyze the properties of the PowerPC unit such as treatment of rounding modes, precision and error accumulation, treatment of exceptional cases and Java floating point. I designed the micro-architecture and implemented a fully pipelined floating point unit using Module Compiler from Synopsys. I also implemented an extensive PLI to verify the unit.

Graduate Technical Intern, Intel Corp., Austin, TX
Summer 2000-Fall 2000
Design of a VLIW Instruction Set Architecture for multimedia applications. Analysis of multimedia kernels for performance on high performance processors with multimedia extensions. I reported directly to an Intel Fellow/VP investigating a strategic research project in stream processing. I single handedly designed the instruction set and architecture for a VLIW stream processor directly competing with a vector processor architect with three decades of industry experience and a stream processor research group at Intel's Microprocessor research labs in collaboration with Stanford. Based on characterization information (area, power, delay) for Intel's IA-64 architecture, I developed a small VLIW streaming architecture that consumed 10% of the area and power of the main core but delivered 10 times the performance of the main core on streaming tasks thus easily meeting the design goal provided to me. I also guided the work of an Intel engineer who developed an architecture simulator based on my design.

Software Engineer, Novell Inc. R&D Center, Bangalore, India

1995-1997
Design and implementation of TCP based packet burst protocol and broadcast mechanism as Unix System V kernel modules to improve the performance of network file I/O. Implementation of a multi-threaded Unix kernel driver for file sharing across multiple address spaces to accelerate burst reads and writes. Strategy to multi-thread the core engine of the NetWare NOS on several Unix platforms. Researched and prototyped a new product to do enterprise wide multi-platform user management. Implemented a multi-platform prototype network information service daemon. Guided the development of an IP firewall for Linux based on a Forth kernel built into the OS.

Technical Intern, Center for Development of Imaging Technology, India
Summer 1994
Design and simulation of analog filter circuits for the Indian space program (ISRO) using the Mentor Graphics system.

Skills

- VLSI design with Verilog, Verilog PLI programming.
- Power estimation and modeling using Spice and simulators.
- Design and implementation of micro-architecture simulators.
- FPGA based prototyping.
- Tools:
  - Synopsis Module Compiler, VCS, Design Analyser, Nanosim, Power compiler, Silicon Ensemble.
  - HSpice, ADFMI models, dc_shell scripting.
- Programming languages: C, C++, Python, Lisp, MIPS, ARM, and x86 Assembly language.
- Network programming using sockets, TLI, STREAMS, RPC.
- OS design and implementation: Industry experience programming the SVR4 Unix kernel and experience in porting and modifying the Linux and FreeBSD kernels. Exposed to a large variety of Unix variants. Experience in implementing an x86 protected mode micro-kernel OS from scratch.

Invited Talks

- December 2004, “XStream: CAD tools for Rapid Generation of Embedded Processors”, Siemens, Berkeley.
- July 2003, “Streaming Architectures for Perception”, CVA Group Seminar, Stanford University.
- November 2000, “Augmented von Neumann Processors”, ASPLOS IX, WCI Session, Boston.
- January 2000, “Design of a Parallel Vector Access Unit for SDRAM Memory Systems”, High Performance Computing Group Seminar, Department d’Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona.

Professional Activities

- Program committee member for Memory performance: Dealing with Applications , Systems and Architecture Workshop (MEDEA) 2003.
- Reviewer for HPCA 2005, Async 2004, MEDEA 2003, PACT 2002, ASPLOS 2002, HPCA 2000 and Grace Hopper 2000, IEEE Transactions on Computers, IEEE Transactions on Systems, Man and Cybernetics, IEE, Journal of Embedded Computing.

Public Domain Code

- Nlambda: An embedded processor simulator and Linux distribution.
  `http://nlambda.sourceforge.net`
- Defun/Defmacro/Repl: Verilog implementation of embedded processors.
- CGEN: A symbolic layout generator for static CMOS circuits.
- Visual feature recognizer.
- WormKit: Software infrastructure for fault tolerant distributed applications.
  `http://www.cs.utah.edu/\~mbinu/code/`

Academic Projects Supervised

- Gesture Recognizer. Undergraduate senior project, School of Computing, University of Utah, 2004. Supervised Scott DuVall, Shane Head and Kyle Anderson who developed a gesture recognition system that processed in real-time, hand gestures made by a user in front of a camera. The system was successfully used to control slide navigation for a presentation.
- Face Recognizer. Undergraduate senior project, School of Computing, University of Utah, 2003. Supervised Robert Evans, Robert Smith and Justin Elliss, who designed a face recognition system that accepted images from a camera in real-time, detected faces in the image frames and recognized individuals. This project was selected by the faculty as the best undergraduate project of 2003.
- Packet Filter. Undergraduate senior project, Birla Institute of Technology and Novell IDC, 1995. Guided Amarnath Ramakrishnan who developed a network packet filter for the Linux kernel based on dynamic code generation.

Other Academic Projects

MPP Interconnect Network: Designed a scalable MPP interconnect architecture including network topology, network interface, routing and congestion control protocols and deadlock avoidance/detection mechanism. Designed a prototype circuit based on asynchronous macro-modules.
Wormhole Router: Used ViewLogic to design a Worm-hole router for an MPP interconnect using Actel FPGAs.
Lisp to Asynchronous Circuit Compiler: Designed and implemented a compiler that translates circuits described as concurrent processes in a Lisp like language to a macro-modular asynchronous circuit.
Symbolic Layout Generation for Static CMOS Circuits: Designed and implemented a tool to do technology mapping and symbolic layout generation for multi level combinational circuits by CMOS cell generation.
Micro-kernel OS: Designed and implemented from scratch, a micro-kernel OS for the Intel 80386 and later processors. (B.Tech Thesis Project)
WormKit: A three member group implemented the WormKit, a library that provides the infrastructure for building fault tolerant wormed applications. I did the overall design and implemented the threaded interpretive language and directory services that formed the core of WormKit.

Publications

All papers are available online at `http://www.cs.utah.edu/\~mbinu/pubs/`.

- A Low Power Architecture for Embedded Perception, BINU K. MATHEW, AL DAVIS AND MICHAEL A. PARKER, *International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES), 2004.*
- A Loop Accelerator for Low Power Embedded VLIW Processors, BINU K. MATHEW, AL DAVIS, *International Conference on Hardware/Software Codesign and System Synthesis (CODES + ISSS), 2004.*

- A Low-Power Accelerator for the SPHINX 3 Speech Recognition System, BINU K. MATHEW, AL DAVIS AND ZHEN FANG, *International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES), 2003.*
- Perception Coprocessors for Embedded Systems, BINU K. MATHEW, AL DAVIS AND ALI IBRAHIM, *Workshop on Embedded Systems for Real-Time Multimedia (ESTIMedia), 2003.*
- A Characterization of Visual Feature Recognition, BINU K. MATHEW, AL DAVIS AND ROBERT EVANS, *IEEE 6th Annual Workshop on Workload Characterization, 2003.*
- The Impulse Memory Controller, LIXIN ZHANG, ZHEN FANG, MICHAEL A. PARKER, BINU K. MATHEW, LAMBERT SCHAELICKE, JOHN B. CARTER, WILSON C. HSIEH, AND SALLY A. MCKEE, *IEEE Transactions on Computers, Special Issue on Advances in High Performance Memory Systems, 2001*.
- Design of a Parallel Vector Access Unit for SDRAM Memory Systems, BINU K. MATHEW, SALLY A. MCKEE AND JOHN B. CARTER, AL DAVIS, *Proceedings of the Sixth Annual Symposium on High Performance Computer Architecture (HPCA), 2000.*
- Algorithmic Foundations for a Parallel Vector Access Memory System, BINU K. MATHEW, SALLY A. MCKEE AND JOHN B. CARTER, AL DAVIS, *Proceedings of the Eleventh ACM Symposium on Parallel Algorithms and Architectures (SPAA), 2000.*

Book Sections

- Stream Processors and their Application to the Wireless Domain, BINU K. MATHEW AND ALI IBRAHIM, in *Computer Engineering Handbook, Second Edition, CRC Press LLC, 2006.*
- Very Large Instruction Word Architectures (Updated), BINU K. MATHEW, in *Computer Engineering Handbook, Second Edition, CRC Press LLC, 2006.*
- Very Large Instruction Word Architectures, BINU K. MATHEW, in *Computer Engineering Handbook, First Edition, CRC Press LLC, 2001.*

Work in Progress

- Stream Processors, BINU K. MATHEW, in *A technical book I am currently writing for CRC Press LLC, Expected 2007*
- Time vs Space Multiplexing

Current Research: Rapid SoC Design Using Customized Processors (Siemens)

My current research project named XStream focuses on the rapid design of complex ASICs and SoCs based on the use of highly efficient custom stream processors as primary building blocks. The technology is being developed by a three member team lead by me at the Siemens Technology to Business Center in Berkeley, CA. Core ideas are being patented at the moment. These will form the foundation for a new company spun off from Siemens named Satva Design Automation (`http://www.satvad.com`).

Increasing CMOS integration has lead to the emergence of system on chips (SoCs) that combine complex hardware and software subsystems. This ever-increasing complexity leads to escalating design costs that are compounded by faster time-to-market pressures. Having reached the limits of conventional hardware-oriented Electronic Design Automation (EDA) tools, complex SoC designers and architects are transitioning to the Electronic Systems Level (ESL) methodology that leverages both hardware and software to implement complex SoCs. XStream is a perfect example of a tool that can greatly boost productivity using a mixed hardware/software approach to system design.

This research addresses issues such as: expressing design trade-offs in terms that make sense at the application level, performing benchmark-driven exploration, automatic derivation of extremely lightweight processors or ensembles of processors that are optimized for the particular problem at hand, automatic generation of a software tool chain

and simulation infrastructure, generating accurate energy and area estimates, generating designs that are correct by construction, thereby greatly speeding up system verification and timing closure, automatically generating interfaces to connect newly generated modules to existing IP modules and exploring the trade-off between programmability and hardware efficiency

Post-doctoral Research: Stream Processing (Stanford)

Stream processors are well suited for well structured repetitive applications in the media, signal processing, wireless communications and perceptual application domains that require high computational throughput and communication bandwidth simultaneously with good area and power efficiency. My current research in stream processing addresses three key areas. a) Energy-delay optimization and architecture exploration for embedded low power stream processors. b) Algorithms that integrate simultaneous task scheduling and on-chip memory allocation for stream processors. c) Time vs Space multiplexing: Traditionally, DSP architectures have favored a pipeline of processor style design where multiple processors are chained together with the output of algorithms running on one processor fed as input to algorithms running on the next processor. This is called space multiplexing. Scientific applications typically use a time-multiplexed approach where all processors work on one algorithm in parallel, then move on to the next algorithm and so on. My research analyzes the area, power and compute efficiencies of both styles for a diverse set of applications.

Ph.D. Research: Perception Processing (Utah)

Computers of the near future need to efficiently process perception-oriented workloads like large vocabulary speech recognition and computer vision. Early estimates show that the computation requirements of such workloads will exceed 10GOPS. Even if the high end computers of tomorrow can solve these problems, by their very nature, perception tasks are more useful on low-end and mobile platforms ranging from PDAs, automobile computers and information kiosks, to gadgets embedded into automated homes and offices. The power and performance requirements of perception algorithms are orders of magnitude beyond the capabilities of typical embedded processors. My research focuses on the generation of domain-specific stream processors that can accelerate the performance of a variety of perception algorithms at low power budgets. Over a set of perception and streaming benchmarks, my prototype delivers 1.75 times the performance of a 2.4 GHz Pentium 4 while using only 1/15th of the energy consumed by an Intel XScale embedded processor. This corresponds to a factor of 135 improvement in the energy delay product when compared to a state of the art embedded processor. Details are available online at `http://www.cs.utah.edu/\~mbinu/research/`.

MS Thesis Research: Parallel Vector Access Memory System (Utah)

Base-stride vectors are common in scientific code. I developed a new mathematical technique to decompose a base-stride vector into multiple vectors that can be accessed in parallel on a multi-bank memory system. This approach was implemented in the hardware for an SDRAM memory controller. This memory system achieved speedups ranging from 4 to 39 times for a variety of scientific kernels.

Graduate Coursework

Advanced Computer Architecture
Parallel Computer Architecture
Design and Evaluation of Advanced Computer Architectures
Advanced Digital VLSI Systems Design
Switching Theory (Classic CAD Algorithms)
VLSI Architecture
Seminars:
High Performance I/O Architecture
Networking

CAD of Digital Circuits
Fundamentals of Integrated Circuit Design
Hardware Emulation
Operating Systems
Advanced Operating Systems
Programming Languages

High Performance Memory systems
OS and Compilers

References: Provided on request