بسم الله الرحمن الرحيم

Soha S. Zaghloul Abdalla Mekki

PhD, Computer Science

Assistant Professor

Educational Background

**PhD in Computer Science**, Cairo University, Egypt, January 2006.

**MSc in Computer Science,** American University in Cairo, February 1999.

**Intermediate Diploma in Computer Science,** American University in Cairo, School of Continuing Education, July 1987.

**BSc in Chemical Engineering,** Cairo University, July 1983.

Academic Posts

**Assistant Professor** at King Saud University (KSU) – Sep 2011 till present

**Assistant Professor** at Misr International University (MIU) – Jan 2008 till Aug 2011.

**Teaching Assistant** The American University in Cairo (AUC), School of Sciences and Engineering – Feb 1999 till Jun 2003

**Instructor** at the American University in Cairo, the School of Continuing Education (SCE), Sep 1990 till Dec 2007

Teaching Employment History

**King Saud University – Faculty of Computer and Information Science**

* Taught the following courses:
* CSC 422: Networks II
* CSC 111: Java Programming I
* CSC 524: Advanced Computer Networks
* CSC 528: Interconnection Networks
* Supervised the following Master Projects:
* Quantitative Comparative Study of Bare-Metal and Hosted Hypervisors.
* Parallelizing a Frequency-Domain Watermarking Algorithm on a Multicore Computer
* Testing the Mutual Effect of Virtualization and Parallelism in a Cloud Environment
* The Implementation of Live Migration Using CloudSim
* Supervised the following Undergraduate Projects:
* Parallel Sudoku
* iDiet – Diet Planner System on iPad

**Misr International University – Faculty of Computer Science**

* Taught the following courses:
* Computer Organization
* Digital Logic Design
* Computer Networks
* Operating Systems
* Compiler Design
* Data Structure
* Professional Ethics
* Supervised a graduation project entitled ***“Simulation of a Simplified Cisco Router”***

**The American University in Cairo – School of Continuing Education**

* Cisco Certified Network Administrator (CCNA)
  + Train the instructors on both the practical and theoretical parts of all four levels of CCNA.
  + Certify trainers.
  + Teach undergraduate and postgraduate students; train them in all four levels of the CCNA, practically and theoretically.
* Taught the following courses:
  + Pascal
  + C Language
  + Data Structure
  + Discrete Maths
  + Computer Organization
  + Other introductory courses

**The American University in Cairo - School of Science & Engineering**

* Assisted in teaching the following graduate courses:
  + Contemporary Computer Design
  + Parallel Computer Architecture
  + Digital Image Processing and Pattern Recognition
  + Neural Networks and Genetic Algorithms
* Assisted in teaching the following undergraduate courses:
  + C Language
  + Algorithms

SEMINARS

* “The Mutual Effect of Virtualization and Parallelism in a Cloud Environment” within the activities of the Software Knowledge Engineering Research Group (SKERG), KSU, Oct 2013.
* “A Technique for Load-Balanced Management of Security Tasks in Grids” within the activities of the SKERG, KSU, Oct 2013.
* “Partitioned Services Layer Autonomous System for Cloud Computing”, within the activities of the SKERG, KSU, Oct 2013.
* “The Mutual Effect of Virtualization and Parallelism”, IEEE Africon Conference, Sep 2013.
* “Watch the Cloud”- within the activities of the Communications and Networking Research Group (CNRG), KSU, March 2012.

Researches

* “Evaluation of OpenMP Parallel Programs Performance on Virtual Machines”, funded by RC, KSU.

Industrial Employment History

***September 2002 till December 2007***

***Senior Coordinator for Cisco Programs – Full Time***

**The American University in Cairo – School of Continuing Education (SCE)**

**Initially, I set the Cisco network lab from scratch. Then, my responsibilities included the following tasks:**

* Maintained the Cisco network lab
* Specified the required equipment for the new certificates
* Estimated the required budget for the proposed certifications
* Recommended potential providers
* Attend Cisco conferences and exhibitions and submit a relevant report
* Explored new Cisco certificates to adopt at the SCE
* Followed up the instructors working for the Cisco program
* Hire *Cisco local academies* to work under the umbrella of the *Cisco Regional Academy* (AUC)
* Follow up the performance of the hired *Cisco local academies*
* Ensure the smooth running of the Cisco program



***September 1990 till August 2002***

***Technical Support – Full Time***

**The American University in Cairo – School of Continuing Education**

**My responsibilities included the following tasks:**

* Assist students in conducting their assignments
* Advised students
* Maintained computer labs



***September 1989 to September 1990***

***Technical Support – Full Time***

**SARHANK Microsoft Authorized Dealer**

**My responsibilities included the following tasks:**

* Report about newly devised software imported from Microsoft
* Conduct demonstrations and training sessions on the new software



***October 1986 till September 1989***

***Technical Support – Full Time***

**The American University in Cairo – School of Continuing Education**

**My responsibilities included the following tasks:**

* Assisted students in conducting their assignments
* Maintained computer labs

Academic Projects and Researches

**Soha Zaghloul, Muhamed Mudawar and Gamal Darwish. Development of a Simulatneously Threaded Multi-Core Processor. In Proceedings of the ITI/IEEE 3rd International Conference on Information and Communication Technology, December 2005, Cairo, Egypt.** *Abstract* **(1)**



**Soha Zaghloul. Development of a Simultaneously Multithreaded Computer Systems – in fulfillment of the degree of Doctor of Philosophy, Cairo University, Faculty of Information and Computer Science, December 2005.** *Abstract* (2)



**Soha Zaghloul. Backtracking in Wormhole Routing Interconnection Networks – in partial fulfillment of the degree of Masters of Science, The American University in Cairo, Computer Science Department, School of Science and Engineering, February 1999.** *Abstract* (3)



***Efficient Routing Algorithms.*  – Project.** *Abstract* (4)



***Cache Design.*  – Project**

A Cache Memory is implemented using VHDL on UNIX. Different cache organizations and update policies are explored.



***English-Arabic Translator –* Project**

A thorough survey was made about the Arabic grammar and its correspondence to the English one. Although the two languages are not of the same origin, but a simple translator was made using LISP.



***Handwriting Recognition*. – Project**

The project was implemented using *Matlab*.



***Patients Database System.* – Project**

This is a database system maintained by a hospital. The database keeps information about the personal record of its patients as well as detailed information of their medical history. The medical information took into consideration all fields needed by a physician in different specialties. A careful survey was made before implementation amongst physicians of different specializations to specify the relevant information which is important to them. The project was written using ***Rbase***.

Training and Certificates

**Cisco CCNA Training – LTScotland, Glasgow**

***Since September 2002 till May 2003***

* Basics of Internetworking
* Routers and Routing
* Switches
* WAN Technologies



**IBM Authorized Center – Egypt, Cairo**

***July 1997***

* Bridges, Routers and Gateway Technologies



**Advanced Computing Technology (ACT) Novell Authorized Center – Egypt, Cairo**

***Since October 1995 till December 1995***

* Novell: Introduction to Networking
* Novell: Netware 4.1 Administration
* Novell: Netware 4.1 Installation and Configuration

Technical Skills

**Programming Languages and Packages**

* Assembly Language
* Pascal
* C Language
* Visual / Borland C++
* HTML
* Renoir: Hardware Computer-Aided Design Package
* Matlab
* VHDL

PERSONAL SKILLS

* Ability to present sophisticated topics in a simple way to the audience
* Capacity to learn new skills and embrace new ideas
* Ability to work independently or collaboratively in a team
* Ability to seek answers for imposed problems independently
* Attentive to details
* Proactive, motivated and enthusiastic
* Hardworker and dedicated

languages

Arabic: Mother Language

English: Fluent writing and speaking

French: Fluent writing and speaking (French-educated)

ABSTRACTS

***(1) Abstract:***

Simultaneous Multithreading (SMT) becomes one of the major trends in the design of future generations of microarchitectures. Its key strength comes from its ability to exploit both thread-level and instruction-level parallelism; it uses hardware resources efficiently. Nevertheless, SMT has its limitations: contention between threads may cause conflicts; lack of scalability, additional pipeline stages, and inefficient handling of long latency operations. Alternatively, Chip Multiprocessors (CMP) are highly scalable and easy to program. On the other hand, they are expensive and suffer from cache coherence and memory consistency problems.

This research proposes a microarchitecture that exploits parallelism at instruction, thread and processor levels. It merges both concepts of SMT and CMP. Like CMP, multiple cores are used on a single chip. Hardware resources are replicated in each core except for the secondary-level cache which is shared amongst all cores. The processor applies the SMT technique within each core to make full use of available hardware resources. Moreover, the communication overhead is reduced due to the inter-independence between cores.

Results show that the proposed microarchitecture outperforms both SMT and CMP. In addition, resources are more evenly distributed amongst running threads.



***(2) Abstract:***

The main objective of processor designers is to effectively exploit the parallelism available in different applications. Simultaneous Multithreading (SMT) becomes one of the major trends in the design of future generations of microarchitectures. Its key strength comes from its ability to exploit both thread-level and instruction-level parallelism. In addition, its efficient use of hardware resources resolves conflicts raised between parallel instructions and consequently hides latency. Nevertheless, SMT has its limitations: having multiple threads active concurrently may cause conflicts on the hardware resources. Moreover, SMT processors suffer from their lack of scalability, additional pipeline stages, and inefficient handling of long latency operations.

Alternatively, Chip Multiprocessors (CMP) are highly scalable and easy to program. On the other hand, they are expensive due to duplication of resources. Besides, it encounters cache coherence and memory consistency problems.

The proposed microarchitecture exploits parallelism at instruction, thread, and processor levels. It merges both concepts of SMT and CMP. Like CMP, multiple cores are used on a single chip. Hardware resources are replicated in each core except for the secondary-level cache which is shared amongst all cores. In order to reduce the cost, the processor uses these hardware resources as efficient as possible by applying the SMT in each processor. Moreover, the cores are made totally independent in order to reduce the communication overhead.

The used simulator is the *ss\_smt*, which is built on the basis of *SimpleScalar*. It is written in *C* and runs under *Linux*. Many modifications are made in order to adapt the simulator to the new processor’s design. The original simulator issued a single instruction at a time from each module; it is modified so that it issues multiple instructions from each module in each cycle. Multiple applications are loaded in each module and instructions are issued on a round-robin basis. If the instruction that belongs to the thread in turn is not ready, then the next application is checked.

Four benchmarks are used to get the results: they all belong to *SPEC2000*; namely, *mcf*, *gzip*, *vortex*, and *ammp*. The first three benchmarks belong to *SPECint*, whereas the last one belongs to *SPECfp*. All benchmarks are coded in *C* language.

In this research, the experiments used four modules with four applications. Experiments are then repeated with different distribution of threads amongst modules. In the first case, all threads run within a single module: this represents an SMT case. In the second case, each module contains two threads: this is the case of a Chip Multi-Threading (CMT). Finally, each module contained a single thread: this is the case of CMP.

Results show that the proposed microarchitecture outperforms both SMT and CMP. In addition, resources are more evenly distributed amongst running threads; consequently, more efficient use of hardware resources is guaranteed.

More tests are made to estimate the best number of sets and banks in the CMT architecture for all used memory types. An additional experiment is made to test the efficiency of various memory hierarchies. It is shown that the use of a primary-level cache is deemed vital: a drop in the number of the IPC (Instructions per Cycle) occurs if no primary cache is used. The elimination of the secondary-level cache doesn’t have the same impact.

Additionally, the proposed architecture is compared to *SEMPRE* project for various sizes of cache and number of sets. It is found that CMT has always achieved higher IPC as compared to *SEMPRE*. However, the average miss ratios are less in the primary-level instruction and data caches of the *SEMPRE* design. In contrast, the secondary-level unified cache gives better results in CMT.



***(3) Abstract:***

This research is an attempt of devising a new routing algorithm to be implemented in wormhole-switched interconnection networks. In fully adaptive routing, a packet may follow one of many alternative paths in the network. When a packet is emitted at a source node, the packet header looks for an available output buffer. If there is more than one free output channel, how can the header make its decision to select a specific one? In most algorithms, this selection is made randomly: the packet header chooses any free output channel. However, the random selection of an output buffer may lead the packet header to a blocked path after one or more steps. Therefore, the packet is blocked at a certain node, whereas it might have been progressed till its destination if it used another output buffer. The question is therefore how to predict that a specific path will not hinder the progress of a packet?

The concept under research is that of *backtracking* in wormhole-switched networks. Whenever a packet finds itself blocked at a certain node, it returns back and looks for an alternative path instead of waiting for the target path to be released. The packet therefore explores all possible paths to its destination. Although backtracking was previously implemented in circuit switching networks, it was not applied on wormhole-switched networks at the time of implementing this work.



***(4) Abstract:***

The selection of a switching technique, a routing algorithm and a topology is of a major importance in the design of an interconnection network. Nowadays, the wormhole routing technique is given a special importance; it is expected to be the one prevailing in the next generations. Although switching techniques affect the network performance heavily, the routing algorithms are not of less importance. The latter should be efficient enough, deadlock-free, livelock-free, and starvation-free as well. In addition, the routing algorithms should be fault-tolerant so that the network does not completely drop in case of the failure of one of its nodes and/or links. Furthermore, the topology of a network has a great impact on its performance. One of the most common topologies used currently is the *k-ary n-cube*. In fact, this topology embeds many other classes of topologies such as meshes, rings, tori, and hypercubes. This explains the popularity of this topology.

This research is subdivided into two main parts: ***Routing in Wormhole Networks*** amd ***Efficient Routing Algorithms in k-ary n-cube Systems***. In addition, a simple routing simulator is devised. The simulator is written in *C* language.