AzTE Arizona Technology Enterprises

www.azte.com

# An Efficient Stack Data Management for Scratchpad Memory Based Multi-core Processors

**AzTE Case #M13-123P**


## Inventors

***Aviral Shrivastava***

*Associate Professor*
*School of Computing, Informatics, and Decision Systems Engineering*

***Ke Bai***

*Graduate Research Assistant*
*School of Computing, Informatics, and Decision Systems Engineering*

***Jing Lu***

*Graduate Research Assistant*
*School of Computing, Informatics, and Decision Systems Engineering*


## Intellectual Property Status:

*Patent Pending*

## Background

As technology scales, the number of cores in processors will rapidly increase in order to avoid the power wall. In processors that have hundreds and thousands of cores, traditional memory architectures – in which a coherent memory interface is provided to all the cores in hardware, will not be feasible. The memory management functionality is expected to migrate from hardware (in existing systems) to the software domain. As a result, existing programs – that have been written assuming that processor hardware will provide memory management support – will not execute correctly on these modern many-core architectures. To make existing programs run correctly on such modern many-core architectures, programmers will have to manually change the program by inserting memory management functionality in the program. However, this is extremely difficult, and even if possible, manual memory management is tedious and a highly error-prone process

## Invention Description

Researchers at Arizona State University have developed a compiler that can automatically compile a given application to execute on a scratchpad memory based multi-core architecture. A novel stack data management technique comprises a modified compiler infrastructure and optimized stack data management runtime library. The compiler can manage limitless stack data, solve invalid pointers, perform stack management with smaller footprint on local memory, and scale with the number of cores. Benchmark results ran on an IBM Cell processor show overhead of only 3.23% in performance.

## Potential Applications

- **Chip Manufacturers** – reduce programming overhead of implementing scratchpad memory based multi-core processors.
- **Application Developers** – create motivation to efficiently utilize the multitude of multi-core processors available in the market.

## Benefits and Advantages

- **Automated** – compiler can automatically compile applications for scratchpad memory on multi-core architecture.
- **Unlimited Stack Data in Local Memory** – the compiler can request dynamic memory allocation in the global memory and support unlimited stack data in the local memory.
- **Better Stack Pointer Management** – pointer problems caused by pointing to other stack frames in global memory will no longer become an issue.
- **Reduced Number of Direct Memory Access (DMA)** – stack frames are managed at a coarser granularity.
- **Improved Performance** – stack data is managed in linear fashion, which reduces the management overhead on maintaining memory status if memory is fragmented by circular fashion. Therefore, our management overhead is only 3% on average.

## Contact


*Bill Loux*

Director of Business Development, Physical Sciences

Arizona Technology Enterprises, LLC (AzTE)

P: 480.884.1992

F: 480.884.1984

BLOUX@AZTE.COM

TECHNOLOGYVENTURES@AZTE.COM