A Multithreaded Processor Architecture for Executing Functional Programs

Eiji Itoh

School of Information Science,
Japan Advanced Institute of Science and Technology

February 14, 1997

Keywords: Multithreaded Processor Architecture, Functional Programs, Hazards .

1 Introduction

Progress of the integrated circuit microscopic fabrication technologies have reduced the switching time of MOS devices. However, the wiring delay time does not shorten, it is difficult to improve microprocessor clock speed.

To overcome the above difficulty, one of the most efficient method is the division of a pipeline stage into some shorter stages. The shorter pipeline stages by division, the smaller wiring delay time gets, and microprocessor clock speed is improved.

However, if pipeline stages of traditional microprocessors that execute a single thread are divided, pipeline execution is stalled by instruction dependencies in a single executing thread.

On the other hand, if pipeline stages of multithreaded microprocessors are divided, the pipeline utilization is increased since no instruction dependencies among multiple threads are exist.

As mentioned above, multithreaded processor architectures would be a right direction for progress of the integrated circuit microscopic fabrication technologies.

This paper proposes a high performance multithreaded processor architecture for executing functional programs, and describes an outline of a design implementing the processor architecture and evaluates performance based on behavior logic synthesis.

2 Functional Programs and Multithreaded Processor Architecture

Functional programs include parallelism and any terms of their expressions may be evaluate concurrently. Thus a functional program can be converted multiple execution threads, while a multithreaded processor requires a lot of threads to exhibit intrinsic performance. Hence the ability of multithreaded processors can be brought out by rich parallelism of functional programs.

3 MUP : Multithreaded Ultra-pipeline Processor architecture

A multithreaded processor which executes functional programs, make the pipeline actions very efficient as follows;

1. Filling up each pipeline stage with a instruction issued by different threads. Each instruction on a pipeline stage does not depend on each other. Hence data hazards and branch hazards are avoided. The processor can executes the instructions at the rate of a CPI of 1.0. And the pipeline control logics can be reduced.

2. Increasing the depth of pipelining. By dividing a pipeline stage into several minuter stage, processor clock can be speeded up, or the pipeline throughput is increased.

A *Multithreaded Ultra-pipeline Processor architecture(MUP)* proposed in this paper takes advantage of efficient pipelining above mentioned. The MUP has following features;

Own hardware resources for each thread: The MUP fills each pipeline stage with a instruction issued by distinct thread. Since the MUP deal with the same number of the threads as the number of pipeline stages, each thread must be assigned his own hardware resources. Hence structure hazards are avoided.

A thread-ID pipeline: The pipelines of the MUP are classified into the following two categories;

- *Instruction execution pipeline.* Through this pipeline, instructions are executed .
- *Thread-ID pipeline.* It keeps the index number to identify a thread. It indicates which thread instruction executed on the current pipeline stage.

A thread select unit chooses a thread to be executed next cycle: In order to fill each pipeline stage with a instruction issued by distinct thread, the alternation of thread at every clock cycle is performed according to round-robin scheduling policy. Consequently, instructions which are executed in the same thread is executed strictly in turn, and data hazards and branch hazards are avoided.

Two split caches: The MUP has a instruction cache and a data cache, and so structure hazards are avoided.

Pipelined caches and pipelined register files: To provide the high throughput for the processor, cache memories and register files are pipelined.

A shared large sized cache by all thread: To reduce the cache miss rate and to improve cache miss usage, the caches with sufficient capacity for all thread are shared by all of them.

Since all threads shares the caches and the MUP alters the execution thread at every clock cycle, these would be no possibility of cache access locality. However if the caches size is in proportion the number of threads, the performance holds satisfactorily.

No stalling the pipeline: The MUP does not stall the pipeline to keep high throughput even if a thread causes a cache miss. No stalling pipeline mechanism is included in a cache miss processing unit.

Therefore the MUP avoids data hazards, branch hazards, and structure hazards. The MUP achieves high throughput by high speed clock rate.

4 Design and Performance Evaluation

4.1 Design

A MUP prototype is designed and synthesized by hardware behavior description. This prototype has the following features;

Instruction set: Instruction set of the MUP consists of general instructions except integer multiplication, integer division, and floating point arithmetic.

Hardware configuration: The MUP consists of a thread select unit, a set of program-counters, an instruction cache, a data cache, a control unit dealing with cache miss, a set of register-files, a sign extend unit, a shift unit, a execution unit, a address calculate unit, and an exception handling unit.

Pipeline configuration: The pipeline pitch of the MUP is decided under consideration of access time of the cache memory cell array. Supposing the access time of the cache memory cell array is 1/it nsec, the pipeline of the MUP is decided to 17 stages.

4.2 Performance Evaluation

In order to examine the multithreaded ultra-pipeline processor architecture proposed here, the prototype design is evaluated. The design is written in a behavior description language. The description is converted to net list by logic synthesis tool. Then, the operating clock frequency and the amount of hardware are calculated. The result is shown as follows;

- the number of gates : 48,705 (5,059 flip-flops are contained in this number)
- the number of bits of register file and others : 20,228
- the operating clock frequency : $1\,GHz$

5 Conclusion

This paper the first proposes the multithreaded ultra-pipeline processor architecture for executing functional programs. The MUP has sufficient hardware resources for each thread, the thread select unit pick out a thread to be executed next cycle, and the two split caches. Therefore the processor avoids data hazards, branch hazards, and structure hazards. To dealing with multiple threads, the MUP has the thread-ID pipeline in addition to.

The second, the MUP prototype has been designed to examine the amount of gates and the operating clock frequency. The MUP prototype has been logically synthesized at gate level. As the result, the number of gates are 48k, the register file and others require 20k bits. And it is shown that the operating clock frequency of MUP is $1\,GHz$. According by, the multithreaded ultra-pipeline processor architecture is a right direction progress of integrated circuit microscopic fabrication technologies.