A Multi-threaded ATM Switching Architecture

Rieko Matsuda

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

February 13, 1998

Keywords: Multi-thread, ATM, Software processing .

Abstract

1 Introduction

ATM(Asynchronous Transfer Mode) is regarded as a transmission and switching technology which required for a implementation of B-ISDN(Broadband Integrated Digital Network). It is one of the important technological issues that construcs an appropriate switch architecture for ATM, and various ATM switch architectures have been proposed.

This paper proposes "A Multi-threaded ATM Switching Architecture" as a way of implementation of ATM switches. This ATM switch is a shared memory typed. The Multi-threaded ATM Switch executes cell swicthing by the software processing on the multi-threaded processor, and the software processing provides the switch with flexible to in bandwidth allocation.

The feature of Multi-threaded processor is that a thread has a hardware context". Therefore, a pipelined processor can switch threads per clock without overhead of the context switch. To apply the feature to such a ATM switch, resource allocations per clock is possible, each thread gets access to one shared memory in time division, so there is no contention in the memory access, and it enables flexible resource allocations to various virtual circuits. In this paper, each cell switching procedure runs as a thread on the processor. The swicthing threads have no dependency each other, therefore processing resources can be used efficiently, consequently throughput increases.

This paper states that cell switching processsing with the multi-threaded processor is more effective than with the single-threaded processor which has only one hardware context.

2 System Design of Multi-threaded ATM Switch

This section gives the details of the Multi-threaded ATM Switch. Suppose that cell switching is carried out under the following specifications.

Multi-threaded ATM Switch Specification

- Speed × Input/Output line number : 155 Mbps × 8 lines
- Buffering method : shared memory
- Switch processor : Multi-threaded Processor based on MIPS architecture
- Switch processor clock frequency : 1 GHz
- Bandwidth to shared memory : at least 340Mbyte/s
- Input/Output part : putting I/O processor

I/O Processor Specification

- Only one I/O processor Read/Write cell from/to shared memory
- Status indication : input flag and output flag
- processing throughput : 6 Mcell/s

Cell Switching Algorithm

Cell switching is done by eight running cell switching threads on the switch processor. Each of the eight threads has charge of one pair of input line and output line.

The cell switching algorithm is briefly specified below.

1. **I/O processor :** I/O processor writes one input cell into the shared memory.
2. **Each of threads on switch processor:** Each of threads reads a header of the input cell from the shared memory, change it by looking at the translation table, and put the cell into output queue which is applicable to output line and service category. Each of threads decides an output cell according to the output scheduling algorithm, write its address in the command register of the I/O processor.
3. **I/O processor :** I/O processor reads the output cell.

3 Performance Evaluate Simulation

This section presents the verification of the ATM switch design in the previous section whether it is enough for the target processing performance.

The verification is carried out with a simulator that executes MIPS instruction set programs. The simulator runs the cell switching program, and outputs the total execution instruction steps.

The following list gives the matters of verification .

- Validness of software processing with multi-threaded processor
- Comparison to software processing with single-threaded processor
- Performance in case of eight running threads on multi-threaded processor which has an eight-stage instruction pipeline.
- Performance in case of processor synchronization support.

It supposes that MIPS processor runs at frequency of 1GHz, as described previous section, Assuming that it takes one clock cycle to execute one instruction, it takes 1[ns] to execute one instruction. Consequently, this simulation is able to estimate the execution time from instruction counts which each thread executes.

4 Results of Simulation and Considerations

First, the validness of software processing with multi-threaded processor is described. From the switch specification, the switch have to process eight cells within 2730[ns]. In case only one thread runs, the maximum number of instructions required to process one cell is 183 instructions. Therefore, it takes 1464[ns] to switch eight cells. Consequently, there is enough time to do more instructions processing additional procedure, and the validness of software processing is susceptible of proof.

Secondly, the comparison to software processing with single-threaded processor is described. In the single-threaded processor, the maximum number of instructions per one cell is 247 instructions, and this case is the same condition as that the maximum number occur in the multi-threaded processor. Therefore, in single-threaded processor, it takes 1.35 times as long time as in the multi-threaded processor. That is to say, it is 35 % more efficiently in the multi-threaded processor than in the single-threade processor. The factor of increasing of instructions execution time grows in proportion to dependencies between instructions and overheads of context switch.

Thirdly, the performance in case of eight running threads on multi-threaded processor which has an eight-stage instruction pipeline is described. In the best case, it is 83 % more efficiently in the multi-threaded processor than in the single-threaded processor. Therefore, the performance improvement to the single-threaded processor which has an eight-stage pipeline more grows than which has a five-stage pipeline.

Finaly, the performance in case of processor synchronization support is described. This synchronization support is implemented by using spinlock. The overheads of synchronization is at most 30% in case of getting lock without spin.

5 Conclusion

A Multi-threaded ATM Switching Architecture which proposed in this paper is the architecture to which the features of ATM exchange and a multi-threaded processor applied. Such a switching architecture will be expected to increase throughput and provide flexibility. This paper described that the verification about the validness of the proposed switching architecture.

The most important matter have gotten through this simulation is stated following.

- In the multi-threaded processor, the more the number of pipeline-stage, the more Performance improvement to the single-threaded processor increases.
- Performance in the multi-threaded processor which supports synchronization does not lower than in the single-threaded processor, because the synchronization support is not so large overhead.

Therefore, Software processing on the multi-threaded processor more increases throughput, and can be more efficiently than software processing on the single-threaded processor. In future, another issue to provide flexibility by software processing will be examined.