A Pipelined Cache Mechanism for a Multithreaded Processor

Aihara Kouichi

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

February 14, 1997

Keywords: multithread,pipeline,throughput,cache memory.

1 Introduction

Several multithreaded processor architectures have been proposed to improve performance of pipelined processors. The clock cycle time or the processor throughput depends on the cache performance on the highly pipelined processors.

Since the multithreaded processor deals with multiple instruction stream the cache capacity must be balanced with the number of threads. The cache capacity is the larger, the longer the access time of it is, because the wiring delay time becomes longer too.

First, this paper proposes a pipelined cache mechanism with high throughput, and a partitioning of cache memory cell array to reduce memory access time.

Second, the cache hit rate and throughput of a multithreaded processor are evaluated and several problems concerning to those are discussed.

2 A Pipelined Cache Mechanism

The cache throughput for a multithreaded processor is more important because the machine cycle time of it depends on the cache cycle time rather than the cache access time or the cache latency. Hence, a pipelined cache mechanism for high throughput is proposed.

The cache access is carried out through the following processes: the address decode, the read data from a memory cell array, the judgment for hit or miss in cache, and the data out. A four stage pipelined cache memory is constructed by assigning each one of those operations to a pipeline stage.

Multithreaded processor needs the large cache memory capacity because multiple threads access the cache memory for every cycle. Therefore, designers of the cache memories introduce a memory cell array partitioning method and a hierarchical address decoding scheme. By memory cell array partitioning, the wiring delay time is shorten. Furthermore, the decoder can decode address and select the memory cell block simultaneously. Hence, both the high speed memory access and the larger cache capacity can be achieved at a time.

Combining with hierachical address decoding, memory cell array partitioning and pipelined cache mechanism, a high throughput cache memory can be organized.

3 Cache Memory Architecture and The Memory Organization

It is said that block conflict would be frequently occured in a cache memory of a multithreaded processor.

If a thread has a private cache or a cache dedicated to itself, the data or instructions are mapped only on itself. While, if all thread share a cache having the sufficient capacity for the number of all threads, there is no restriction of the cache mapping. Therefore, the shared cache may provide sufficient number of blocks for each one of threads.

However, even if sufficient number of blocks are provided, the block conflict may occur between threads. In this case, the set associative mapping method can reduce conflict occurrence.

Even if a cache miss stops progress of one thread, another thread continues to require access for the cache. Therefore, the access of a main memory occurs very frequently. Moreover, there is a very large difference between the processor cycle time and the main memory access time. If the frequence of main memory request exceeds the throughput of the main memory, a queue of the main memory access grows.

It is effective to adopt the write-back strategy when a write operation to the memory occurs. The write-back scheme reduces the number of access requests to the main memory. And deviding the main memory into multiple units and providing a access queue for each unit, the length of the access queue can be shortened. The aim of those schemes is lowering the memory cycle utilization rate.

The reduction of the time to renew a block where a cache miss occurs is concerned with the inprovement of throughput. To sum up the above discussion, the cache entry strategy have great influence on the processor throughput.

4 Simulation of Cache

To show the effectiveness of the schemes above mentioned, the cache hit rate and the processor throughput are investigated by the simulation.

The throughput is defined as numbers of the transaction data to numbers of the execution clock cycles The throughput is a substantial measure indicating the system

performance.

5 Consideration

From the simulation results of the cache hit rate when direct map method is used, it is obvious that the hit rate of a private cache is higher than the hit rate of a shared cache. There may be more block conflict in the shared cache than the private one, because the block conflict occurs among threads.

However, addition of set associativity in the seared cache has an effect of increase of the cache hit rate. By a cache has a set associativity, the cache can map the block in the same line. But the increase of hit rate is not extended for increasing set associativity.

As the memory access penalty increases, the throughput falls down abruptly. Because if miss penalty is larger, the condition of growing queue is realized.

When write-through strategy is used, the throughput falls down strikingly because the frequency of the memory request are remarkably large. The programs are almost mapped in a instruction cache, then the memory accesses are dominated by the occurrence store of instructions. Write-through strategy must do a write request in every store instruction, so the memory request queue grows rapidly.

To prevent such situation, it is necessary to reduce numbers of the memory request. From this point of view, the write-back is a effective cache entry method.

Furthermore, if the memory are composed with multiple units and a memory request queue is provided for each unit, the queue length can be reduced, because the memory requests are distributed among memory banks, and memory request queue for each unit is greatly reduced. Consequently, this scheme prevents from fall down of the throughput.

6 Conclusion

First, this paper proposed a pipelined cache mechanism for a multithreaded processor. Even if the pipelined cache mechanism increases in cache latency, it can be improved the processor throughput. It is possible to close the pipelined pitch and increase capacity of the cache memory at a time by means of a partitioning of cache memory cell array to reduce memory access time. The increase of the cache capacity can be improved the hit rate. As it can be reduced penalty of a cache miss, the processor throughput is improved.

Second, this paper proposed the set associative mapping method to reduce the block conflict and the write-back strategy to reduce the number of access requests to the main memory. Even if the cache mechanism is complicated by those, and so increases in cache latency, a lowering of throughput is prevented by the more pipelined cache mechanism.

Therefore, a pipelined cache mechanism achieves an important role in to improve the processor throughput.

Through the organization of a cache memory and the simulation in this paper, the processor performance is not rightly evaluated only the cache hit rate. So, it is obvious that the cache hit rate is high comparatively, however, the throughput is not high.

To improve performance of a multithreaded processor, it is important for the length of memory request queue to reduce. Consequently, the memory performance is very important.

Because the processor speeds continue to increase faster than the main memory access times, the main memory will increasingly be a factor that limits performance. Consequently, the key of a high performance of global system is to extend band width of the main memory by deviding the main memory into multiple units.