Memory Dependence Prediction

ECE 721: Advanced Microarchitecture
Department of Electrical and Computer Engineering
North Carolina State University

Rhushikesh Prabhune **Ajinkya Tundurwar**

May 1, 2020

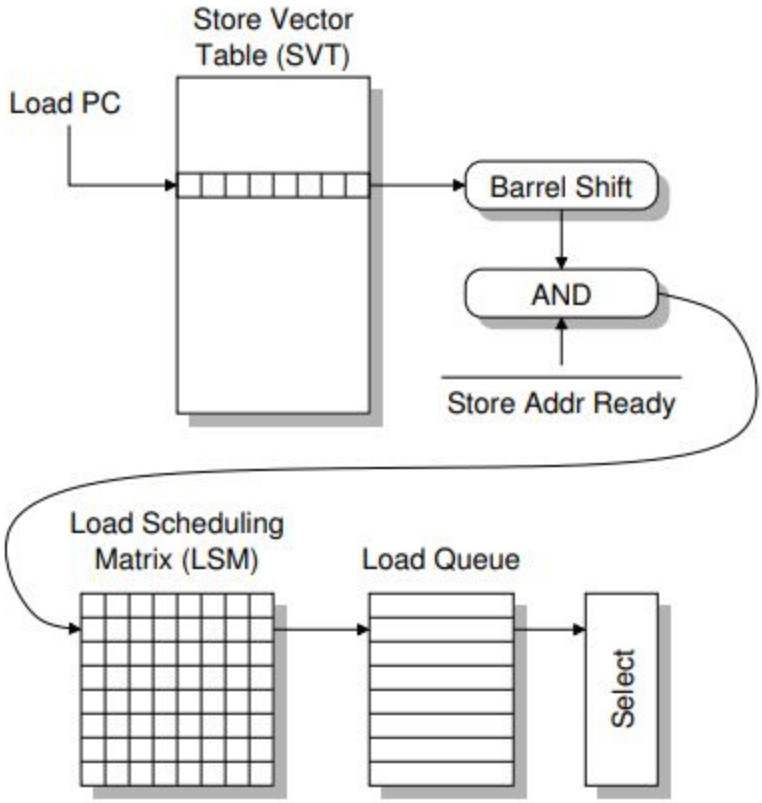
Related Work

Memory Dependence Prediction is a useful technique used to predict load-store dependency and thereby improve performance by correctly stalling/unstalling the loads. By letting execution of loads which are predicted to be not dependent on any store, the overall parallelism of the superscalar processor is improved and thus, IPC performance is also improved.

Loads can be stalled for any store with unknown address (i.e, non-speculative execution of load) or the load can be assumed to have zero dependencies and thus cleared for execution (i.e, blind speculation of load). These two extremes can be useful for certain benchmarks depending on the number of load-store instructions in the benchmark. To improve the IPC performance, a memory dependence prediction unit can be deployed in 721sim or any other simulator for making a decision whether a particular load needs to be stalled or not. The Sticky-Bit, Store Sets are some of the algorithms which can be deployed in the simulator for reducing memory-order violations. The Store Vector Memory Dependence Predictor is another such predictor, described by Samantika Subramaniam and Gabriel H. Loh, which can be used for predicting load-store dependencies.

Store Vector Memory Dependence Predictor:

Store Vectors predictor uses distance between the load's most recent store and the violating store to specify precisely which stores cause a violation. The store vectors use two data structures, namely, Store Vector Table (SVT) and Load Scheduling Matrix (LSM).



The load accesses the SVT in the dispatch stage to get the load's corresponding store vector table entry. The least significant bit of the store vector table entry gives information about the dependency of the load with the most recent store. The SVT entry is then passed through a barrel shifter, which rotates the SVT vector such that the LSB of the vector is aligned with the most recent store. The load will then wait for already resolved dependencies if the bits corresponding to these dependencies are not cleared which could result in a deadlock. The unwanted bits are cleared using a bitwise AND of the store vector with the bits from each store queue entry which indicates if the store address is available or not.
The store vector is then inserted into the LSM after processing for scheduling. The LSM is a matrix with rows equal to the load queue size and columns equal to the store queue size. The stores, once executed, clear the corresponding columns related to itself. A wired NOR tells whether any unresolved predicted dependencies remain or not. When there are no dependencies, the wired NOR is set to 1, and the load can be sent for execution.

The vectors in the SVT are all initialised to zero. Therefore, initially all the loads will undergo blind speculation. After a load store violation occurs, the store's relative age with respect to the load is determined. The bit corresponding to this relative age is then set in the load's SVT.
The SVT vector may have all the bits set, after which the processor will behave in a non-speculative manner. Therefore, the vectors need to be reset periodically to keep the speculative functionality of the processor. The resetting mechanism can be similar to the sticky bit mechanism where the bits are periodically clear after a particular number of instructions have been retired.

References

- G. Z. Chrysos and J. S. Emer, ”Memory dependence prediction using store sets”.
- Samantika Subramaniam and G. H. Loh, ”Store vectors for scalable memory dependence prediction and scheduling”.
- A. Moshovos, ”Memory Dependence Prediction,” Ph.D dissertation, Computer Sciences Department, University of Wisconsin - Madison, USA, 1998.