

CLAIMS

1. A method for processing an instruction sequence that includes a barrier instruction, a load instruction preceding the barrier instruction, and a subsequent memory access instruction following the barrier instruction, the method comprising:
 - initiating the load instruction;
 - in response to initiating the load instruction, initiating the barrier instruction;
 - initiating the subsequent memory access instruction following initiating the barrier instruction;
 - determining, by a processor core, that the load instruction is resolved based upon receipt by the processor core of: an earliest of a good combined response for a read operation corresponding to the load instruction; and data for the load instruction;
 - in response to the load instruction being resolved, completing the barrier instruction;
 - determining whether execution of the subsequent memory access instruction is initiated prior to completion of the barrier instruction;
 - if execution of the subsequent memory access instruction is not initiated prior to completion of the barrier instruction, initiating by the processor core, in response to determining the barrier instruction completed, execution of the subsequent memory access instruction; and
 - if execution of the subsequent memory access instruction is initiated prior to completion of the barrier instruction: in response to determining the barrier instruction not completed, tracking of the subsequent memory access instruction with respect to invalidation; and, discontinuing by the processor core, in response to determining the barrier instruction completed, tracking of the subsequent memory access instruction with respect to invalidation.
2. The method of claim 1, further comprising:
 - inserting, by the processor core, an entry for the load instruction into a load reorder queue; and
 - marking, by the processor core, the entry in the load reorder queue as resolved in response to determining, by the processor core, that the load instruction is resolved.

3. The method of either of the preceding claims, wherein the subsequent memory access instruction is a subsequent load instruction.
4. The method of any of claims 1 to 2, wherein the subsequent memory access instruction is a subsequent store instruction.
5. The method of any of the preceding claims, further comprising the processor core receiving the good combined response for the read operation corresponding to the load instruction prior to receiving the data for the load instruction.
6. The method of any of claims 1 to 4, further comprising the processor core receiving the good combined response for the read operation corresponding to the load instruction subsequent to receiving the data for the load instruction.
7. The method of any of the preceding claims, wherein the barrier instruction completes when all load and store instructions prior to the barrier instruction are resolved.
8. A data processing system configured to process an instruction sequence that includes a barrier instruction, a load instruction preceding the barrier instruction, and a subsequent memory access instruction following the barrier instruction, the data processing system comprising:
 - a cache memory; and
 - a processor core coupled to the cache memory, the system comprising means for:
 - initiating the load instruction;
 - in response to initiating the load instruction, initiating the barrier instruction;
 - initiating the subsequent memory access instruction following initiating the barrier instruction;
 - determining, by a processor core, that the load instruction is resolved based upon receipt by the processor core of: an earliest of a good combined response for a read operation corresponding to the load instruction; and data for the load instruction;
 - in response to the load instruction being resolved, completing the barrier instruction;

determining whether execution of the subsequent memory access instruction is initiated prior to completion of the barrier instruction;

if execution of the subsequent memory access instruction is not initiated prior to completion of the barrier instruction, initiating by the processor core, in response to determining the barrier instruction completed, execution of the subsequent memory access instruction; and

if execution of the subsequent memory access instruction is initiated prior to completion of the barrier instruction: in response to determining the barrier instruction not completed, tracking of the subsequent memory access instruction with respect to invalidation; and,

discontinuing by the processor core, in response to determining the barrier instruction completed, tracking of the subsequent memory access instruction with respect to invalidation.

9. The data processing system of claim 8, wherein the system is further operable for:

inserting an entry for the load instruction into a load reorder queue; and

marking the entry in the load reorder queue as resolved in response to determining that the load instruction is resolved.

10. The data processing system of either of claims 8 or 9, wherein the subsequent memory access instruction is a subsequent load instruction.

11. The data processing system of either of claims 8 or 9, wherein the subsequent memory access instruction is a subsequent store instruction.

12. The data processing system of any of claims 8 to 11, further operable for the processor core receiving the good combined response for the read operation corresponding to the load instruction prior to receiving the data for the load instruction.

13. The data processing system of any of claims 8 to 11, further operable for the processor core receiving the good combined response for the read operation corresponding to the load instruction subsequent to receiving the data for the load instruction.
14. The data processing system of any of claims 8 to 13, further operable for the barrier instruction completing when all load and store instructions prior to the barrier instruction are resolved.
15. The data processing system of any of claims 8 to 14, wherein the cache memory is a level two cache memory.
16. The data processing system of claim 15, wherein the processor core includes a load-store unit that includes a load reorder queue (LRQ) and LRQ operable for resolving logic, and wherein the LRQ resolving logic is operable for:
 - inserting an entry for the load instruction into the LRQ; and
 - marking the entry in the LRQ as resolved in response to determining that the load instruction is resolved.
17. A computer program product for processing an instruction sequence that includes a barrier instruction, a load instruction preceding the barrier instruction, and a subsequent memory access instruction following the barrier instruction, the computer program product comprising:
 - a computer readable storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method according to any of claims 1 to 7.
18. A computer program stored on a computer readable medium and loadable into the internal memory of a digital computer, comprising software code portions, when said program is run on a computer, for performing the method of any of claims 1 to 7.

CLAIMS

1. A method for processing an instruction sequence that includes a barrier instruction, a load instruction preceding the barrier instruction, and a subsequent memory access instruction following the barrier instruction, the method comprising:

determining, by a processor core, that the load instruction is resolved based upon receipt by the processor core of an earliest of a good combined response for a read operation corresponding to the load instruction and data for the load instruction;

if execution of the subsequent memory access instruction is not initiated prior to completion of the barrier instruction, initiating by the processor core, in response to determining the barrier instruction completed, execution of the subsequent memory access instruction; and

if execution of the subsequent memory access instruction is initiated prior to completion of the barrier instruction, discontinuing by the processor core, in response to determining the barrier instruction completed, tracking of the subsequent memory access instruction with respect to invalidation.

2. The method of claim 1, further comprising:

inserting, by the processor core, an entry for the load instruction into a load reorder queue; and

marking, by the processor core, the entry in the load reorder queue as resolved in response to determining, by the processor core, that the load instruction is resolved.

3. The method of either of the preceding claims, wherein the subsequent memory access instruction is a subsequent load instruction.

4. The method of any of claims 1 to 2, wherein the subsequent memory access instruction is a subsequent store instruction.

5. The method of any of the preceding claims, further comprising the processor core receiving the good combined response for the read operation corresponding to the load instruction prior to receiving the data for the load instruction.

6. The method of any of claims 1 to 4, further comprising the processor core receiving the good combined response for the read operation corresponding to the load instruction subsequent to receiving the data for the load instruction.

7. The method of any of the preceding claims, wherein the barrier instruction completes when all load and store instructions prior to the barrier instruction are resolved.

8. A data processing system configured to process an instruction sequence that includes a barrier instruction, a load instruction preceding the barrier instruction, and a subsequent memory access instruction following the barrier instruction, the data processing system comprising:

a cache memory; and

a processor core coupled to the cache memory, the system comprising means for: determining that the load instruction is resolved based upon receipt by the processor core of an earliest of a good combined response for a read operation corresponding to the load instruction and data for the load instruction;

if execution of the subsequent memory access instruction is not initiated prior to completion of the barrier instruction, initiating by the processor core, in response to determining the barrier instruction completed, execution of the subsequent memory access instruction; and

if execution of the subsequent memory access instruction is initiated prior to completion of the barrier instruction, discontinuing by the processor core, in response to determining the barrier instruction completed, tracking of the subsequent memory access instruction with respect to invalidation.

9. The data processing system of claim 8, wherein the system is further operable for:

inserting an entry for the load instruction into a load reorder queue; and

marking the entry in the load reorder queue as resolved in response to determining that the load instruction is resolved.

10. The data processing system of either of claims 8 or 9, wherein the subsequent memory access instruction is a subsequent load instruction.

11. The data processing system of either of claims 8 or 9, wherein the subsequent memory access instruction is a subsequent store instruction.

12. The data processing system of any of claims 8 to 11, further operable for the processor core receiving the good combined response for the read operation corresponding to the load instruction prior to receiving the data for the load instruction.

13. The data processing system of any of claims 8 to 11, further operable for the processor core receiving the good combined response for the read operation corresponding to the load instruction subsequent to receiving the data for the load instruction.

14. The data processing system of any of claims 8 to 13, further operable for the barrier instruction completing when all load and store instructions prior to the barrier instruction are resolved.

15. The data processing system of any of claims 8 to 14, wherein the cache memory is a level two cache memory.

16. The data processing system of claim 15, wherein the processor core includes a load-store unit that includes a load reorder queue (LRQ) and LRQ operable for resolving logic, and wherein the LRQ resolving logic is operable for:

inserting an entry for the load instruction into the LRQ; and

marking the entry in the LRQ as resolved in response to determining that the load instruction is resolved.

17. A computer program product for processing an instruction sequence that includes a barrier instruction, a load instruction preceding the barrier instruction, and a subsequent memory access instruction following the barrier instruction, the computer program product comprising:

a computer readable storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method according to any of claims 1 to 7.

18. A computer program stored on a computer readable medium and loadable into the internal memory of a digital computer, comprising software code portions, when said program is run on a computer, for performing the method of any of claims 1 to 7.