

CLAIMS

1. A computer program product for executing an instruction within a computing environment, said 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 comprising:

obtaining, by a processor, a machine instruction for execution, the machine instruction being defined for computer execution according to a computer architecture, the machine instruction comprising:

an operation code to specify a nontransactional store operation;

a first operand; and

a second operand to designate a location for the first operand; and

executing, by the processor, the machine instruction, the executing comprising:

nontransactionally placing the first operand at the location specified by the second operand, wherein information stored at the second operand is retained despite an abort of a transaction associated with the machine instruction, and wherein the nontransactionally placing is delayed until an end of transactional execution mode of the processor.

2. The computer program product of claim 1, wherein the end of transactional execution mode results from an end of an outermost transaction associated with the machine instruction or an abort condition.

3. The computer program product of claim 1, wherein multiple nontransactional stores appear as concurrent stores to other processors.

4. The computer program product of claim 1, wherein the method further comprises:
determining whether the processor is in transactional execution mode;
based on the processor being in the transactional execution mode, determining whether the transaction is a constrained transaction or a nonconstrained transaction; and
based on the transaction being a nonconstrained transaction, continuing execution of the machine instruction.

5. The computer program product of claim 4, wherein based on the transaction being a constrained transaction, providing a program exception and terminating execution of the machine instruction.
6. The computer program product of claim 4, wherein based on the processor not being in transactional execution mode, executing the machine instruction as a store instruction.
7. The computer program product of claim 1, wherein the second operand is formed from a combination of contents of a register specified by an index field of the machine instruction, contents of a register specified by a base field of the machine instruction, and contents of at least one displacement field.
8. The computer program product of claim 1, wherein the machine instruction comprises a nontransactional store instruction, and wherein the method further comprises:
 - initiating the transaction based on execution of a transaction begin instruction, the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;
 - committing the transactional stores to main memory based on executing a transaction end instruction that completes execution of the selected transaction;
 - discarding the transactional stores based on an abort that ends execution of the selected transaction; and
 - wherein the nontransactionally placing is performed independent of the committing or the discarding.
9. The computer program product of claim 1, wherein the first operand comprises contents of a register specified by the machine instruction, the contents being user-specified.
10. The computer program product of claim 1, wherein the transaction is a nonconstrained transaction in a nesting of nonconstrained transactions.

11. A computer system for executing an instruction within a computing environment, said computer system comprising:

a memory; and

a processor in communications with the memory, wherein the computer system is configured to perform a method, said method comprising:

obtaining, by a processor, a machine instruction for execution, the machine instruction being defined for computer execution according to a computer architecture, the machine instruction comprising:

an operation code to specify a nontransactional store operation;

a first operand; and

a second operand to designate a location for the first operand; and

executing, by the processor, the machine instruction, the executing comprising:

nontransactionally placing the first operand at the location specified by the second operand, wherein information stored at the second operand is retained despite an abort of a transaction associated with the machine instruction, and wherein the nontransactionally placing is delayed until an end of transactional execution mode of the processor.

12. The computer system of claim 11, wherein multiple nontransactional stores appear as concurrent stores to other processors.

13. The computer system of claim 11, wherein the method further comprises:

determining whether the processor is in transactional execution mode;

based on the processor being in the transactional execution mode, determining whether the transaction is a constrained transaction or a nonconstrained transaction; and

based on the transaction being a nonconstrained transaction, continuing execution of the machine instruction.

14. The computer system of claim 13, wherein based on the transaction being a constrained transaction, providing a program exception and terminating execution of the machine instruction.

15. The computer system of claim 13, wherein based on the processor not being in transactional execution mode, executing the machine instruction as a store instruction.
16. The computer system of claim 11, wherein the first operand comprises contents of a register specified by the machine instruction, the contents being user-specified, and wherein the second operand is formed from a combination of contents of a register specified by an index field of the machine instruction, contents of a register specified by a base field of the machine instruction, and contents of at least one displacement field.
17. The computer system of claim 11, wherein the transaction is a nonconstrained transaction in a nesting of nonconstrained transactions.
18. A method of executing an instruction within a computing environment, said method comprising:
- obtaining, by a processor, a machine instruction for execution, the machine instruction being defined for computer execution according to a computer architecture, the machine instruction comprising:
 - an operation code to specify a nontransactional store operation;
 - a first operand; and
 - a second operand to designate a location for the first operand; and
 - executing, by the processor, the machine instruction, the executing comprising:
 - nontransactionally placing the first operand at the location specified by the second operand, wherein information stored at the second operand is retained despite an abort of a transaction associated with the machine instruction, and wherein the nontransactionally placing is delayed until an end of transactional execution mode of the processor.
19. The method of claim 18, wherein multiple nontransactional stores appear as concurrent stores to other processors.
20. The method of claim 18, further comprising:
- determining whether the processor is in transactional execution mode;

based on the processor being in the transactional execution mode, determining whether the transaction is a constrained transaction or a nonconstrained transaction; and

based on the transaction being a nonconstrained transaction, continuing execution of the machine instruction.

CLAIMS

1. A computer program for executing an instruction within a computing environment said computer program product comprising a non-transistery computer readable storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method comprising:

obtaining, by a processor, a machine instruction for execution, the machine instruction being defined for computer execution according to a computer architecture, the machine instruction comprising:

an operation code to specify a nontransactional store operation;

a first operand; and

a second operand to designate a location for the first operand; and

executing by the processor, the machine instruction, the executing comprising:

nontransactionally placing the first operand at the location specified by the second operand, wherein information stored at the second operand is retained despite an abort of a transaction associated with the machine instruction, and wherein the nontransactionally placing is delayed until an end of transactional execution of the processor.

2. The computer program product of claim 1, wherein the end of transactional execution mode results from an end of an outermost transaction associated with the machine instruction or an abort condition.

3. The computer program product of claim 1, wherein multiple nontransactional stores appear as concurrent stores to other processors.

4. The computer program product of claim 1, wherein the method further comprises:
determining whether the processor is in transactional execution mode;
based on the processor being in the transactional execution mode, determining whether the transaction is a constrained transaction or a nonconstrained transaction; and
based on the transaction being a nonconstrained transaction; continuing execution of the machine instruction.