CLAIMS

1. A method of controlling execution within a computing environment, the method comprising steps of:

detecting, by a processor, that execution of an instruction stream has aborted;

based on detecting that execution of the instruction stream has aborted and prior to retrying execution of the instruction stream, initiating execution of a machine instruction to indicate to the processor that re-execution of the instruction stream is a retry, the machine instruction being defined for computer execution according to a computer architecture and comprising:

a field to specify a requested assist operation to be performed by the processor;

a value of the field indicating to the processor that re-execution of the instruction stream is a retry and indicating an action is to be taken to facilitate successful re-execution of the instruction stream.

2. The method of claim 1, wherein the machine instruction is a perform processor assist instruction that further includes an operation code to specify a perform processor assist operation, and wherein the field is separate from the operation code and specifies an abort assist operation.

3. The method of claim 1, wherein the field comprises an operation code specifying an abort assist operation.

4. The method of claim 1, wherein the instruction stream is a transaction, the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction.

5. The method of claim 4, wherein the machine instruction further comprises a register field, the register field identifying a register usable by a program to indicate a number of times the transaction has aborted, and wherein the initiating execution comprises providing the number of times to the processor, the processor to take action based on the number of times the transaction has aborted.

6. The method of claim 5, wherein the method further comprises:

based on execution of the machine instruction and a value of the identified register, determining by the processor one or more actions to facilitate successful re-execution of the transaction; and

re-executing the transaction.

7. The method of claim 6, wherein the performing one or more actions comprises setting state of the processor.

8. The method of claim 1, wherein the instruction stream is a transaction, and wherein the method further comprises:

initiating the transaction via a transaction begin instruction; and detecting an abort of the transaction.

9. The method of claim 8, wherein the method further comprises re-executing the transaction, subsequent to execution of the machine instruction.

10. The method of claim 1, wherein the machine instruction further comprises another field, the another field identifying a register usable by a program to indicate a count of a number of times the instruction stream has aborted, and wherein the initiating execution of the machine instruction includes providing the number of times to the processor, the processor to take action based on the number of times the instruction stream has aborted, wherein different actions are taken based on the number of times the instruction stream has aborted, to enable successful re-execution of the aborted instruction stream.

11. The method of claim 1, wherein the action comprises at least one of:

performing random delays within the processor;

delaying processing within the processor based on a number of times the instruction stream has aborted as indicated by the machine instruction;

temporarily suspending out-of-order execution within the processor;

or disabling branching.

12. The method of claim 1, wherein the action to be taken is to restrict processing of the processor or another processor during the retry, the action to be performed being selected based on an abort count associated with the instruction stream, wherein there are a plurality of abort counts configured that have selectable actions associated therewith at different levels of

restrictiveness, and wherein the abort count associated with the instruction stream is an abort count of the plurality of abort counts.

13. The method of claim 1, wherein the action to be performed is selected based on an abort count associated with the instruction stream, wherein a range of abort counts has one or more selectable actions associated therewith, and wherein the action selected performed comprises one or more actions of the one or more selectable actions associated with the range of abort counts comprising the abort count.

14. A system comprising means adapted for carrying out all the steps of the method according to any preceding method claim.

15. A computer program comprising instructions for carrying out all the steps of the method according to any preceding method claim, when said computer program is executed on a computer system.

CLAIMS

1. A method for controlling execution within a computing environment, the method comprising steps of:

subsequent to aborting execution of an instruction stream, 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:

a field to specify a requested assist operation to be performed by the processor; and

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

obtaining, by the processor, a value of the field, the value to indicate to the processor that an action is to be taken to change a mode of operation of the processor to facilitate successful re-execution of the instruction stream.

2. The method of claim 1, wherein the machine instruction is a perform processor assist instruction that further includes an operation code to specify a perform processor assist operation, and wherein the field is separate from the operation code and specifies a transaction abort assist operation.

3. The method of claim 1, wherein the field comprises an operation code specifying a transaction abort assist operation.

4. The method of claim 1, wherein the instruction stream is a transaction, the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction.

5. The method of claim 4, wherein the machine instruction further comprises a register field, the register field identifying a register usable by a program to indicate a number of times the transaction has aborted, and wherein the executing comprises providing the number of times to the processor, the processor to take action based on the number of times the transaction has aborted.

6. The method of claim 4, wherein the method further comprises:

based on execution of the machine instruction and a value of the identified register, determining by the processor one or more actions to facilitate successful re-execution of the transaction; and

re-executing the transaction based on the changed mode of operation.

7. The method of claim 6, wherein the performing one or more actions comprises setting state of the processor.

8. The method of claim 1, wherein the instruction stream is a transaction, and wherein the method further comprises:

initiating the transaction via a transaction begin instruction; and

detecting an abort of the transaction.

9. The method of claim 8, wherein the method further comprises re-executing the transaction, subsequent to execution of the machine instruction based on the changed mode of operation.

10. The method of claim 1, wherein the machine instruction further comprises another field, the another field identifying a register usable by a program to indicate a count of a number of times the instruction stream has aborted, and wherein the executing includes providing the count to the processor, the processor to take action based on the count to enable successful re-execution of the aborted instruction stream.

11. A system comprising means adapted for carrying out all the steps of the method according to any preceding method claim.

12. A computer program comprising instructions for carrying out all the steps of the method according to any preceding method claim, when said computer program is executed on a computer system.

AMENDED CLAIMS
received by the International Bureau on 19 jULY 2013 (19.07.2013)

1. A method for controlling execution within a computing environment, the method comprising steps of:

detecting, by a processor, that execution of an instruction stream has aborted;

based on detecting that execution of the instruction stream has aborted and prior to retrying execution of the instruction stream, initiating execution of a machine instruction to indicate to the processor that re-execution of the instruction stream is a retry, the machine instruction being defined for computer execution according to a computer architecture and comprising a field to specify a requested assist operation to be performed by the processor, a value of the field indicating that re-execution of the instruction stream is a retry and indicating an action is to be taken to facilitate successful re-execution of the instruction stream, the action comprising at least one of: performing random delays within the processor; delaying processing within the processor based on a number of times the instruction stream has aborted as indicated by the machine instruction; temporarily suspending out-of-order execution within the processor; or disabling branching;

based on initiating execution of the machine instruction, performing the action; and

retrying execution of the instruction stream.

2. The method of claim 1, wherein the machine instruction is a perform processor assist instruction that further includes an operation code to specify a perform processor assist operation, and wherein the field is separate from the operation code and specifies an abort assist operation.

3. The method of claim 1, wherein the field comprises an operation code specifying an abort assist operation.

4. The method of claim 1, wherein the instruction stream is a transaction, the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction.

5. The method of claim 4, wherein the machine instruction further comprises a register field, the register field identifying a register usable by a program to indicate a number of times the transaction has aborted, and wherein the initiating execution comprises providing the number of times to the processor, the processor to take action based on the number of times the transaction has aborted.

6. The method of claim 4, wherein the method further comprises:

based on initiating execution of the machine instruction and a value of the identified register, determining by the processor one or more actions to facilitate successful re-execution of the transaction; and

re-executing the transaction.

7. The method of claim 6, wherein the performing one or more actions comprises setting state of the processor.

8. The method of claim 1, wherein the instruction stream is a transaction, and wherein the method further comprises:

initiating the transaction via a transaction begin instruction; and

detecting an abort of the transaction.

9. The method of claim 8, wherein the method further comprises re-executing the transaction, subsequent to execution of the machine instruction.

10. The method of claim 1, wherein the machine instruction further comprises another field, the another field identifying a register usable by a program to indicate a number of times the instruction stream has aborted, and wherein the initiating execution of the machine instruction includes providing the number of times to the processor, the processor to take action based on the number of times the instruction stream has aborted, wherein different actions are taken based on the number of times the instruction stream has aborted.

11. A system comprising means adapted for carrying out all the steps of the method according to any preceding method claim.

12. A computer program comprising instructions for carrying out all the steps of the method according to any preceding method claim, when said computer program is executed on a computer system.