

CLAIMS

1. A computer program product for providing diagnostic information on transaction aborts, 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 comprising:
 - detecting, by a processor, an abort of a transaction, the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;
 - determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and
 - based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the diagnostic information to include an aborted transaction instruction address.
2. The computer program product of claim 1, wherein there are a plurality of types of transaction diagnostic blocks, and the determining comprises checking whether one or more types of transaction diagnostic blocks are to be stored into based on the abort.
3. The computer program product of claim 2, wherein the plurality of types of transaction diagnostic blocks comprises: a program-specified TDB, a program interruption TDB, and an interception TDB.
4. The computer program product of claim 2, wherein the checking indicates a plurality of transaction diagnostic blocks are to be stored into based on the abort.
5. The computer program product of any preceding claim, wherein the transaction diagnostic block further comprises one or more of: a transaction nesting depth, a transaction abort code, a conflict token, one or more program interruption parameters, contents of one or more general purpose registers at time of the abort, and model dependent diagnostic information.

6. The computer program product of claim 5, wherein the transaction block further comprises one or more of: an exception access identification, a data exception code, a program interruption identification, a transaction exception identification, and a breaking event address.
7. The computer program product of claim 5, wherein the method further comprises executing a transaction abort instruction to abort the transaction, the transaction abort instruction specifying the transaction abort code.
8. The computer program product of claim 1, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB.
9. The computer program product of claim 8, wherein the transaction begin instruction initiates a nonconstrained transaction, the nonconstrained transaction being an outermost transaction.
10. The computer program product of claim 9, wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the transaction begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.
11. The computer program product of claim 1, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the determining comprises:
 - checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;
 - checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and
 - checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.

12. A computer system for providing diagnostic information on transaction aborts, the 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:
 - detecting, by a processor, an abort of a transaction, the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;
 - determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and
 - based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the diagnostic information to include an aborted transaction instruction address.
13. The computer system of claim 12, wherein the transaction diagnostic block further comprises one or more of: a transaction nesting depth, a transaction abort code, a conflict token, one or more program interruption parameters, contents of one or more general purpose registers at time of the abort, and model dependent diagnostic information.
14. The computer system of claim 13, wherein the transaction block further comprises one or more of: an exception access identification, a data exception code, a program interruption identification, a transaction exception identification, and a breaking event address.
15. The computer system of claim 13, wherein the method further comprises executing a transaction abort instruction to abort the transaction, the transaction abort instruction specifying the transaction abort code.
16. The computer system of claim 12, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB, the transaction begin instruction initiating a nonconstrained

transaction, the nonconstrained transaction being an outermost transaction, and wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the transaction begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.

17. The computer system of claim 12, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the determining comprises:

- checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;

- checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and

- checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.

18. A method of providing diagnostic information on transaction aborts, the method comprising:

- detecting, by a processor, an abort of a transaction, the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;

- determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and

- based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the diagnostic information to include an aborted transaction instruction address.

19. The method of claim 18, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB, the transaction begin instruction initiating a nonconstrained transaction, the nonconstrained transaction being an outermost transaction, and wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the

transaction begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.

20. The method of claim 18, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the determining comprises:

checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;

checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and

checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.

AMENDED CLAIMS

received by the International Bureau on 1 October 2013 (01.10.2013).

1. A computer program product for providing diagnostic information on transaction aborts, 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 comprising:

detecting, by a processor, an abort of a transaction, the transaction comprising one or more instructions, and the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;

determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and

based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the diagnostic information including an address of an instruction corresponding to the transaction that aborted, the address of the instruction depending on a reason for the abort of the transaction; the reason provided in an abort code, and wherein: based on the abort code having a first value of one or more first values, the diagnostic information including an address of an instruction that was executing when the abort was detected; based on the abort code have a second value of one or more second values, the diagnostic information including an address of an instruction that is past the instruction executing when the abort was detected; and based on the abort code having a third value of one or more third values, the diagnostic information including an address of an instruction that is earlier or later than the instruction executing when the abort was detected.

2. The computer program product of claim 1, wherein there are a plurality of types of transaction diagnostic blocks, and the determining comprises checking whether one or more types of transaction diagnostic blocks are to be stored into based on the abort.

3. The computer program product of claim 2, wherein the plurality of types of transaction diagnostic blocks comprises: a program-specified TDB specified by a transaction begin instruction, a

program interruption TDB to be used for aborts due to an interruption, and an interception TDB to be used for aborts due to an interception.

4. The computer program product of claim 2, wherein the checking indicates a plurality of *transaction diagnostic blocks* are to be stored into based on the abort.

5. The computer program product of any preceding claim, wherein the transaction diagnostic block further comprises one or more of: a transaction nesting depth, a transaction abort code, a conflict token, one or more program interruption parameters, contents of one or more general purpose registers at time of the abort, and model dependent diagnostic information.

6. The computer program product of claim 5, wherein the transaction block further comprises one or more of: an exception access identification, a data exception code, a program interruption identification, a transaction exception identification, and a breaking event address.

7. The computer program product of claim 5, wherein the method further comprises executing a transaction abort instruction to abort the transaction, the transaction abort instruction specifying the transaction abort code.

8. The computer program product of claim 1, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB.

9. The computer program product of claim 8, wherein the transaction begin instruction initiates a nonconstrained transaction, the nonconstrained transaction being an outermost transaction.

10. The computer program product of claim 9, wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the transaction

begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.

11. The computer program product of claim 1, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the storing comprises determining which of the plurality of types of diagnostic blocks are to be stored into, the determining comprising:

checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;

checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and

checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.

12. A computer system for providing diagnostic information on transaction aborts, the 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:

detecting, by a processor, an abort of a transaction, the transaction comprising one or more instructions, and the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;

determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and

based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the diagnostic information including an address of an instruction corresponding to the transaction that aborted, the address of the instruction depending on a reason for the abort of the transaction, the reason provided in an abort code, and wherein: based on the abort code having a first value of one or more first values, the diagnostic information including an address of an instruction that was executing when the abort was detected;

based on the abort code have a second value of one or more second values, the diagnostic information including an address of an instruction that is past the instruction executing when the abort was detected; and based on the abort code having a third value of one or more third values, the diagnostic information including an address of an instruction that is earlier or later than the instruction executing when the abort was detected.

13. The computer system of claim 12, wherein the transaction diagnostic block further comprises one or more of: a transaction nesting depth, a transaction abort code, a conflict token, one or more program interruption parameters, contents of one or more general purpose registers at time of the abort, and model dependent diagnostic information.

14. The computer system of claim 13, wherein the transaction block further comprises one or more of: an exception access identification, a data exception code, a program interruption identification, a transaction exception identification, and a breaking event address.

15. The computer system of claim 13, wherein the method further comprises executing a transaction abort instruction to abort the transaction, the transaction abort instruction specifying the transaction abort code.

16. The computer system of claim 12, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB, the transaction begin instruction initiating a nonconstrained transaction, the nonconstrained transaction being an outermost transaction, and wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the transaction begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.

17. The computer system of claim 12, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the storing comprises determining which of the plurality of types of diagnostic blocks are to be stored into, the determining comprising:
checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;

checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and

checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.

18. A method of providing diagnostic information on transaction aborts, the method comprising:

detecting, by a processor, an abort of a transaction, the transaction comprising one or more instructions, and the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;

determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and

based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the diagnostic information including an address of an instruction corresponding to the transaction that aborted, the address of the instruction depending on a reason for the abort of the transaction, the reason provided in an abort code, and wherein: based on the abort code having a first value of one or more first values, the diagnostic information including an address of an instruction that was executing when the abort was detected; based on the abort code have a second value of one or more second values, the diagnostic information including an address of an instruction that is past the instruction executing when the abort was detected; and based on the abort code having a third value of one or more third values, the diagnostic information including an address of an instruction that is earlier or later than the instruction executing when the abort was detected.

19. The method of claim 18, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB, the transaction begin instruction initiating a nonconstrained transaction, the nonconstrained transaction being an outermost transaction, and wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the transaction begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.

20. The method of claim 18, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the storing comprises determining which of the plurality of types of diagnostic blocks are to be stored into, the determining comprising:

checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;

checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and

checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.

CLAIMS

1. A computer program product for providing diagnostic information on transaction aborts, *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 comprising:

detecting, by a processor, an abort of a transaction, the transaction comprising one or more instructions, and the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;

determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and

based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the storing diagnostic information in the transaction diagnostic block comprising:

determining a reason for the abort of the transaction;

based on the reason for the abort, determining an address of an instruction corresponding to the transaction that aborted, wherein the address depends on the reason for the abort; and

based on determining the address of the instruction corresponding to the transaction that aborted, storing the address of the instruction corresponding to the transaction that aborted in the transaction diagnostic block.

2. The computer program product of claim 1, wherein there are a plurality of types of transaction diagnostic blocks, and the determining whether diagnostic information is to be stored into a transaction diagnostic block comprises checking whether one or more types of transaction diagnostic blocks are to be stored into based on the abort.

3. The computer program product of claim 2, wherein the plurality of types of transaction diagnostic blocks comprises: a program-specified TDB specified by a transaction begin instruction, a program interruption TDB to be used for aborts due to an interruption, and an interception TDB to be used for aborts due to an interception.

4. The computer program product of claim 2, wherein the checking indicates a plurality of transaction diagnostic blocks are to be stored into based on the abort.

5. The computer program product of any preceding claim, wherein the transaction diagnostic block further comprises one or more of: a transaction nesting depth, a transaction abort code, a conflict token, one or more program interruption parameters, contents of one or more general purpose registers at time of the abort, and model dependent diagnostic information.
6. The computer program product of claim 5, wherein the transaction block further comprises one or more of: an exception access identification, a data exception code, a program interruption identification, a transaction exception identification, and a breaking event address.
7. The computer program product of claim 5, wherein the method further comprises executing a transaction abort instruction to abort the transaction, the transaction abort instruction specifying the transaction abort code.
8. The computer program product of claim 1, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB.
9. The computer program product of claim 8, wherein the transaction begin instruction initiates a *nonconstrained transaction*, the *nonconstrained transaction* being an outermost transaction.
10. The computer program product of claim 9, wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the transaction begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.
11. The computer program product of claim 1, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the storing comprises determining which one or more types of transaction diagnostic blocks of the plurality of types of transaction diagnostic blocks are to be stored into, the determining comprising:

checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;

checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and

checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.

12. A computer system for providing diagnostic information on transaction aborts, the 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:

detecting, by a processor, an abort of a transaction, the transaction comprising one or more instructions, and the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;

determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and

based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the storing diagnostic information in the transaction diagnostic block comprising:

determining a reason for the abort of the transaction;

based on the reason for the abort, determining an address of an instruction corresponding to the transaction that aborted, wherein the address depends on the reason for the abort; and

based on determining the address of the instruction corresponding to the transaction that aborted, storing the address of the instruction corresponding to the transaction that aborted in the transaction diagnostic block.

13. The computer system of claim 12, wherein the transaction diagnostic block further comprises one or more of: a transaction nesting depth, a transaction abort code, a conflict token,

one or more program interruption parameters, contents of one or more general purpose registers at time of the abort, and model dependent diagnostic information.

14. The computer system of claim 13, wherein the transaction block further comprises one or more of: an exception access identification, a data exception code, a program interruption identification, a transaction exception identification, and a breaking event address.

15. The computer system of claim 13, wherein the method further comprises executing a transaction abort instruction to abort the transaction, the transaction abort instruction specifying the transaction abort code.

16. The computer system of claim 12, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB, the transaction begin instruction initiating a nonconstrained transaction, the nonconstrained transaction being an outermost transaction, and wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the transaction begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.

17. The computer system of claim 12, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the storing comprises determining which one or more types of transaction diagnostic blocks of the plurality of types of transaction diagnostic blocks are to be stored into, the determining comprising:
checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;

checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and

checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.

18. A method of providing diagnostic information on transaction aborts, the method comprising:

detecting, by a processor, an abort of a transaction, the transaction comprising one or more instructions, and the transaction effectively delaying committing transactional stores to main memory until completion of a selected transaction;

determining, by the processor, whether diagnostic information is to be stored into a transaction diagnostic block (TDB) based on the abort; and

based on the determining indicating diagnostic information is to be stored, storing diagnostic information in the transaction diagnostic block, the storing diagnostic information in the transaction diagnostic block comprising:

determining a reason for the abort of the transaction;

based on the reason for the abort, determining an address of an instruction corresponding to the transaction that aborted, wherein the address depends on the reason for the abort; and

based on determining the address of the instruction corresponding to the transaction that aborted, storing the address of the instruction corresponding to the transaction that aborted in the transaction diagnostic block.

19. The method of claim 18, wherein the determining comprises determining whether a valid transaction diagnostic block address is provided in a transaction begin instruction, wherein based on a valid transaction diagnostic block address being provided, diagnostic information is stored in a program specified TDB, the transaction begin instruction initiating a nonconstrained transaction, the nonconstrained transaction being an outermost transaction, and wherein the determining whether a valid transaction diagnostic block address is provided comprises checking a base field of the transaction begin instruction of the outermost transaction, wherein a nonzero base field indicates a valid transaction diagnostic block address.

20. The method of claim 18, wherein there are a plurality of types of transaction diagnostic blocks, and wherein the storing comprises determining which one or more types of transaction diagnostic blocks of the plurality of types of transaction diagnostic blocks are to be stored into, the determining comprising:

checking whether a valid transaction diagnostic block address is provided in a transaction begin instruction, and based on the checking indicating a valid transaction diagnostic block address, the determining indicating a program specified TDB is to be stored into;

checking whether the abort is due to an interruption, and based on the abort being due to an interruption, the determining indicating a program interruption TDB is to be stored into; and

checking whether the abort is due to an interception condition, and based on the abort being due to the interception condition, the determining indicating an interception TDB is to be stored into.